UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ARNeuro: Mobile Augmented Reality for Craniotomy Planning

Dissertação de Mestrado

Marcel Oliveira Alves

São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Marcel Oliveira Alves

ARNeuro: Mobile Augmented Reality for Craniotomy Planning

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Daniel Oliveira Dantas

São Cristóvão – Sergipe

2018

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

Alves, Marcel Oliveira
A474a ARNeuro: mobile augmented reality for craniotomy planning / Marcel Oliveira Alves ; orientador Daniel Oliveira Dantas. – São Cristóvão, 2018.
93 f.

Dissertação (mestrado em Ciências da Computação) – Universidade Federal de Sergipe, 2018.

1. Ciências da Computação. 2. Aplicativos móveis. 3. Tablet(computadores). 4. Smartphones. 5. Cirurgia. I. Dantas, Daniel Oliveira, orient. II. Título.

CDU 004.42

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do Curso de Mestrado em Ciência da Computação-UFS.
Candidato: MARCEL OLIVEIRA ALVES

Em 30 dias do mês de Julho do ano de dois mil e dezoito, com início às 15h00min, realizou-se no Auditório do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Marcel Oliveira Alves**, que desenvolveu o trabalho intitulado: ***"ARNeuro: Mobile Augmented Reality for Craniotomy Planning"***, sob a orientação do Prof. Dr. Daniel Oliveira Dantas. A Sessão foi presidida pelo Prof. Dr. Daniel Oliveira Dantas (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. Eduardo Oliveira Freire (PROEE/UFS), Prof. Dr. Hendrik Teixeira Macedo (PROCC/UFS) e, em seguida, ao Prof. Dr. David Correa Martins Junior(UFABC). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO "*(aprovado/reprovado)*" COM "*(com/sem)*" ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária "*Prof. José Aloísio de Campos*", 30 de Julho de 2018.

Prof. Dr. Daniel Oliveira Dantas
(PROCC/UFS)
Presidente

Prof. Dr. Eduardo Oliveira Freire
(PROEE/UFS)
Examinador Interno

Prof. Dr. Hendrik Teixeira Macedo
(PROCC / UFS)
Examinador Interno

Prof. Dr. David Correa Martins Junior
(UFABC)
Examinador Externo

Marcel Oliveira Alves
Marcel Oliveira Alves
Candidato

I dedicate this thesis to my brother Daniel Oliveira Alves

Acknowledgements

Thanking everyone who collaborated with a project is a difficult task. First, understanding the importance of indirect support is hard. Second, choosing the right words to express all the gratitude is impossible. Finally, mentioning every single person who helped me in this road is something that I will certainly fail. In spite of that, this is a task that I must do, and if I miss any name, I hope to fix my mistake with one beer or two.

My first thanks goes to the Creators of the simulation that we live. This piece of flawed (but perfect) work, with all its intricate combinations of events, has brought me to this exact point in time and space, and to the conclusion of this work of my own.

I would like to thank my supervisor, Daniel Dantas, who guided me through this path and offered valuable insight to solve some of the problems I have faced. I also thank the neurosurgeon Bruno Fernandes for his great ideas, feedback, and help in this project. I am also really grateful for the observations of my friend Lucas Xavier, and for the useful tips of the professors Leila Silva, Eduardo Freire, and Leonardo Matos.

The mentioned people have directly contributed to my work, but I also appreciate all the indirect support. For this, I thank my girlfriend, Fabrícia Vicente, for all the motivation, ideas, and, especially, patience, during all this time. I know it was hard when I told her that I would spend Valentine's with my dissertation instead of with her, but she always understood my goals, and even motivated me when everything seemed too complicated. I found a voodoo doll that looked like me in her bag, but I think that it is not going to be needed anymore.

My family was also fundamental in all this process. Many thanks to my mother, Kátia, who gave me emotional support, even though sometimes it was harder for her than for me, and to my father, Washington, who gave me great advice that helped me in keeping my sanity. I do not forget my younger brother, Henrique, who, of course, did not help me at all, but may I serve as an inspiration to him - call me if you ever read this, Henrique. I also thank my uncles Ernani, Roberto, Wanderley, and Sérgio, my aunts Suzy, Lisette, Sandra, and Lúcia, my cousins Patrícia, Gabriel, Isabela, Rafael, Matheus, and Rebecca, my goddaughter, Isadora, and her mother, Suzana, and my grandparents Fátima, Batista, and Zely, because they not only motivated me, but also contributed a lot to my education. Those who are no longer with us, my brother Daniel and my grandfather José, should not be forgotten, because they were of great importance in the foundation of my personality.

Pursuing a MSc degree is no easy task, and reconciling this with a job is even harder. Were not my excellent colleagues, this would be impossible. For this, I thank my MPT coworkers, Heli, Gislene, Vítor, and Creuza, who gave me more support than what I asked for. I also thank the prosecutors Albérico Neves, and Emerson Resende, for believing in me and granting me personal leaves when I needed. I do not forget all the help from the awesome HR guys, Antônio Menezes, and Edson dos Santos. I also thank the prosecutor Raymundo Ribeiro, who was a great mentor for me when I started my job. I am very grateful for the time I spent interning at YouTube, and for this I am really thankful to Fernando Cardoso for giving me that opportunity.

Finally, I would like to thank Felipe Fontes, Milena Chagas, Elisa Menendez, Bárbara Pereira, Alice Cabral, Thuany Andrade, Wedla Melo, João Victor Vasconcelos, Monica Battaglia, Lucas Nabuco, Fábio Barros, Daniel Barros, Diorane Araujo, Gabriel Góis, Francisco Ferreira, Samuel Benson, Gustavo Rocha, Glauber Lima, Ítalo Lessa, Guilherme Cardoso, Ismael Macedo, Irina Jacob, Márcio Dantas, Elias Góis, Douglas Camara, Tural Mehdiyev, Luis Castro, and Tom Golden, for the great friendship and support during this time.

“Mistakes” is the word you’re too embarrassed to use. You ought not to be. You’re a product of a trillion of them. Evolution forged the entirety of sentient life on this planet using only one tool: the mistake.
(Robert Ford, Westworld)

Resumo

Cirurgias minimamente invasivas tornaram-se importantes porque propiciam vários benefícios para o paciente, tais como redução do risco de infecção, menos dor, menor tempo de internação e recuperação mais rápida. Na neurocirurgia, a orientação por imagens é fundamental para a realização de uma cirurgia minimamente invasiva. Por esse motivo, os sistemas de neuronavegação tornaram-se cruciais para os neurocirurgiões. Enquanto as técnicas tradicionais para localização de craniotomia podem apresentar erros de mais de 10mm, os neuronavegadores proporcionam maior precisão, com erros menores do que 5mm.

Apesar de seu papel crucial nos procedimentos neurocirúrgicos, muitos cirurgiões afirmam que a usabilidade dos neuronavegadores tem que ser melhorada. Outros mencionam que os neuronavegadores são caros e inacessíveis em hospitais com poucos recursos. Nesse contexto, alguns pesquisadores sugeriram que soluções de realidade aumentada poderiam ser uma alternativa aos neuronavegadores. Os sistemas de realidade aumentada estão surgindo na neurocirurgia, alguns deles tentando fornecer melhor usabilidade do que os sistemas atuais de neuronavegação, e outros tentando fornecer uma solução mais barata e que possa ser facilmente adotada.

Neste trabalho, analisamos os avanços dos sistemas de realidade aumentadade de neuronavegação nos últimos anos. Percebeu-se que muitas soluções de realidade aumentada superam algumas das deficiências do neuronavegador tradicional, mas também trazem novos problemas. Essas abordagens de realidade aumentada podem ser caras e de difícil adoção, ou exigem muitas etapas pré-operatórias por parte do neurocirurgião, tornando-as indesejáveis para uso em casos reais.

Para resolver esses problemas, apresentamos um aplicativo para dispositivos móveis, chamado ARNeuro, que auxilia na localização da craniotomia. O ARNeuro é mais acessível e mais fácil de usar do que a maioria das abordagens de realidade aumentada. Além disso, o ARNeuro é um aplicativo independente, o que significa que não é necessário nenhum hardware ou software adicional, além do que já está disponível nos hospitais. Nossa solução utiliza Odometria Inercial Visual, através de um *framework* de realidade aumentada, para indicar o centro da região da incisão, sobreposta na cabeça do paciente, mostrada na tela do dispositivo. Nossos experimentos iniciais indicam que o ARNeuro é uma ferramenta promissora. A versão atual do aplicativo tem um erro médio geral menor do que 3,1mm. Apesar disso, o sistema ainda tem muitas limitações e sua acurácia pode ser melhorada.

Palavras-chave: *Smartphone*, *Tablet*, Aplicativo, Neurocirurgia, Cirurgia Minimamente Invasiva.

Abstract

Minimally invasive surgeries have become important because they bring many benefits to the patient, such as reduced risk of infection, less pain, shorter hospital stays, and faster recoveries. In neurosurgery, image-guidance is fundamental to perform a minimally invasive surgery. For this reason, neuronavigation systems became crucial for neurosurgeons. While traditional techniques for craniotomy localization may have a target registration error of more than 10mm, neuronavigators provide higher accuracy, with errors of less than 5mm.

In spite of their crucial role in neurosurgical procedures, many surgeons claim that neuronavigators usability has to be improved. Others mention that neuronavigators are expensive and not accessible in hospitals with few resources. In this context, some researchers suggested that augmented reality solutions could be an alternative to neuronavigators. Augmented reality systems are emerging in neurosurgery, some of them trying to provide better usability than current neuronavigation systems, and others trying to provide an affordable solution that could be easily adopted.

In this work, we analyze the advances of augmented reality neuronavigation systems in the last years. It was noticed that many augmented reality solutions do overcome some of the traditional neuronavigator shortcomings, but they also bring new problems. These augmented reality approaches may be expensive and hard to adopt, or require many preoperative steps from the neurosurgeon, making them undesirable for using in real cases.

To address these issues, we present a mobile application, called ARNeuro, that assists in craniotomy localization. ARNeuro is more affordable and easier to use than most augmented reality approaches. Also, ARNeuro is a standalone application, which means that no additional hardware or software is required, besides what is already available at the hospitals. ARNeuro makes use of Visual Inertial Odometry, provided by an augmented reality framework, to draw the center of the incision region, superimposed on the patient's head, shown on the device screen. Our initial experiments indicate that ARNeuro is a promising tool. The current version of the application has an overall mean target registration error of less than 3.1mm. In spite of that, the system still have many limitations, and its accuracy can be improved.

Keywords: Smartphone, Tablet, App, Neurosurgery, Minimally invasive surgery.

List of Figures

List of Tables

List of source codes

List of abbreviations and acronyms

2D	Two-dimensional
3D	Three-dimensional
AR	Augmented reality
ARRP	Augmented reality registration point(s)
CPU	Central processing unit
CT	Computed tomography
CV	Computer Vision
FRE	Fiducial registration error
GPS	Global Positioning System
GPU	Graphics processing unit
HMD	Head-mounted display
HSV	Hue saturation value
IARC	International Agency for Research on Cancer
IEEE	Institute of Electrical and Electronics Engineers
IGS	Image-guided surgery
IGNS	Image-guided neurosurgery
IM	Image marker(s)
IMU	Inertial measurement unit
LFD	Light Field Display
MD	Doctor of Medicine
MR	Magnetic resonance
MRI	Magnetic resonance imaging
MSc	Master of Science
OR	Operating room
PC	Personal computer
RANSAC	Random sample consensus

RGBA	Red green blue alpha
RMSE	Root-mean-square error
PRP	Physical registration points
PTP	Physical target point
SCS	Screen coordinate system
SD	Standard deviation
SLAM	Simultaneous localization and mapping
SVD	Singular value decomposition
TRE	Target registration error
VIO	Visual inertial odometry
VRP	Virtual registration point(s)
VO	Visual odometry
VTP	Virtual target point
WCS	World coordinate system

List of symbols

cm	centimeter
Hz	Hertz
mm	millimeter
$\in$	Is member of
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers

Contents

1 Introduction

The International Agency for Research on Cancer (IARC) estimated that there were 256,213 new cases of brain and nervous system cancer worldwide in 2012 (FERLAY et al., 2015). In Brazil, the expected number of cases in 2018 is 11,320 (INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA, 2017). In the United States, the five-year relative survival rate for the most aggressive form of primary malignant brain tumor, *glioblastoma multiforme*, is 19% for adults aged between 20 and 44, and 5% for adults aged between 55 and 64 (AMERICAN CANCER SOCIETY, INC., 2017). For these reasons, effective treatment is necessary. In this work, we propose a novel approach, using a mobile augmented reality application, to assist neurosurgeons in craniotomy planning.

1.1 Minimally invasive surgery and image-guidance

Minimally invasive surgery, in which the surgeon performs the smallest possible incision that still provides adequate exposure, has become important due to its benefits, like reduced risk of infection, less pain, shorter hospital stays, and faster recoveries (LÉGER et al., 2017; XIA et al., 2018).

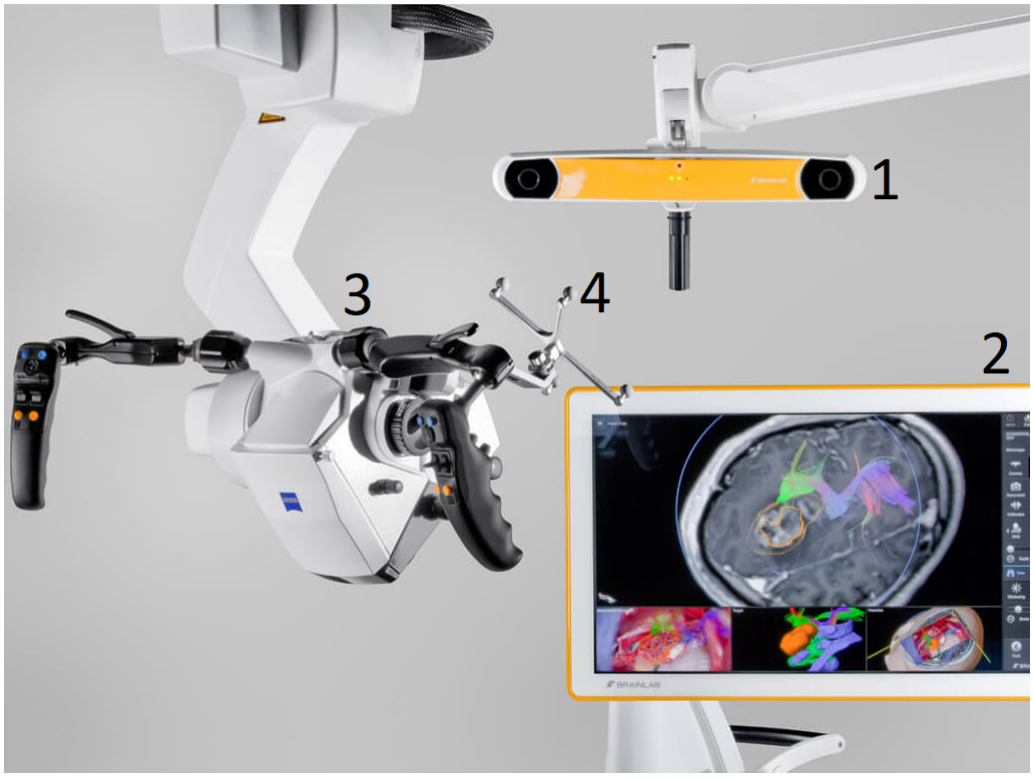
One of the most important requirements for minimally invasive surgeries is the precise localization of the lesion (CHANG et al., 2012; MANDEL et al., 2013; HOU et al., 2016; CHEN et al., 2017), because it allows the surgeon to appropriately tailor the skin incision and reduce the craniotomy size. For this task, image guidance can be of great help. Image-guided surgery (IGS) has improved neurosurgical procedures, bringing minimal invasiveness, safety, and precision (PALEOLOGOS et al., 2000; FRIGHETTO et al., 2003; TABRIZI; MAHVASH, 2015; WATANABE et al., 2016; MEOLA et al., 2017; CUTOLO et al., 2017). A study conducted by Paleologos et al. (2000) shows that IGS was superior than standard surgery in craniotomy procedures to treat meningiomas, because IGS reduced the patients' surgical complications, hospital stay, and costs.

1.2 Neuronavigation

In neurosurgery, the surgeon can perform an image-guided neurosurgery (IGNS) with the assistance of a neuronavigation system, also called neuronavigator. The system main hardware is composed by: (a) an optical tracking device, which tracks the the patient's head and some medical instruments; (b) a monitor, which displays virtual images of the patient's head, usually obtained from a computed tomography (CT) or a magnetic

resonance imaging (MRI) scan. Some neuronavigators also allow the integration of a surgical microscope. To perform the tracking, the instruments have markers attached to them, and the patient's head is immobilized, with a tracked reference frame attached to it. This way, the neuronavigator can locate the medical instruments and the patient's head using the optical tracking device. Figure 1 shows an overview of the system.

Figure 1 – Neuronavigation system with a surgical microscope. 1) Optical tracker; 2) Monitor display; 3) Surgical microscope; 4) Tracking markers attached to the surgical microscope.



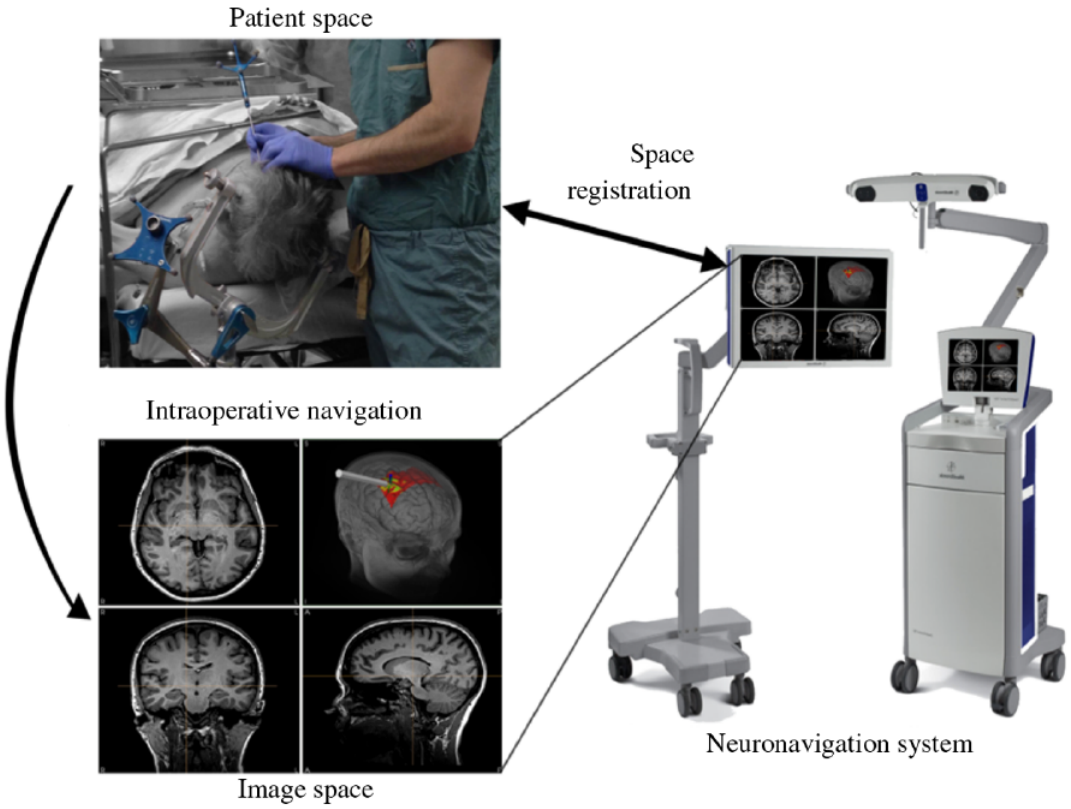
Source – BrainLAB AG (2018) - Modified to include the numbers

In order to make the navigation possible, the surgeon needs to perform a registration step. There are many types of registration (MASCOTT et al., 2006), but the basic idea is that some points marked on the patient's head have to be matched with the correspondent points in the virtual image, creating a map from one space to another. After the registration is complete, when a tracked device touches a position on the patient's head, the correspondent location is highlighted on the virtual image displayed on the monitor. Figure 2 depicts the registration step.

The benefits provided by IGNS turned the neuronavigator into a valuable tool in the operating room (OR) (SCHROEDER et al., 2001; MACIUNAS, 2006; KOCKRO et al., 2009; INOUE et al., 2013; GERARD et al., 2015; HOU et al., 2016; EFTEKHAR, 2016; DROUIN et al., 2017). The results presented by Spivak & Pirouzmand (2005) indicate that neuronavigation systems are much more effective in localizing the brain lesions than conventional techniques.

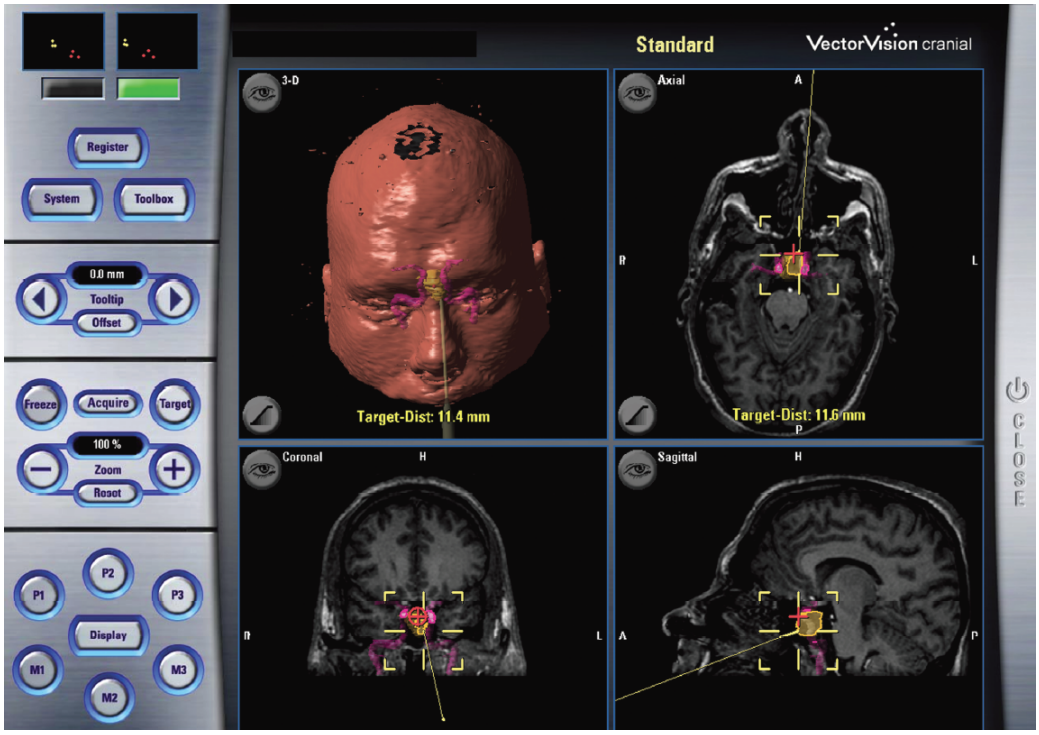
In spite of the advantages they provide, common commercial neuronavigators, such as Medtronic's and BrainLab's, also have limitations. Some authors argue that the usability of monitor-based neuronavigation systems can be improved. For example, Figure 3 shows a screenshot of a neuronavigator monitor. This type of view requires that the neurosurgeon create a "mental map" from the virtual images to the patient (DROUIN et al., 2017). Abhari et al. (2012) state that this limitation causes high cognitive load and slows the process of planning, which can introduce errors. Tabrizi & Mahvash (2015) add that the spatial reasoning requires

Figure 2 – **Patient space**: The patient head is immobilized, with a tracked reference frame attached. The surgeon holds a tracked instrument. **Image space**: Virtual images of the patient displayed by the neuronavigator monitor. **Space registration**: Mapping between *patient space* and *image space*.



Source – Gerard et al. (2018)

Figure 3 – Screenshot of a neuronavigation monitor. The tumor is outlined in yellow and the internal carotid arteries in pink. The red cross is the targeted anterior border of the tumor. The yellow cross and line represent the position indicated by the tracked medical instrument.



Source – Enchev (2009)

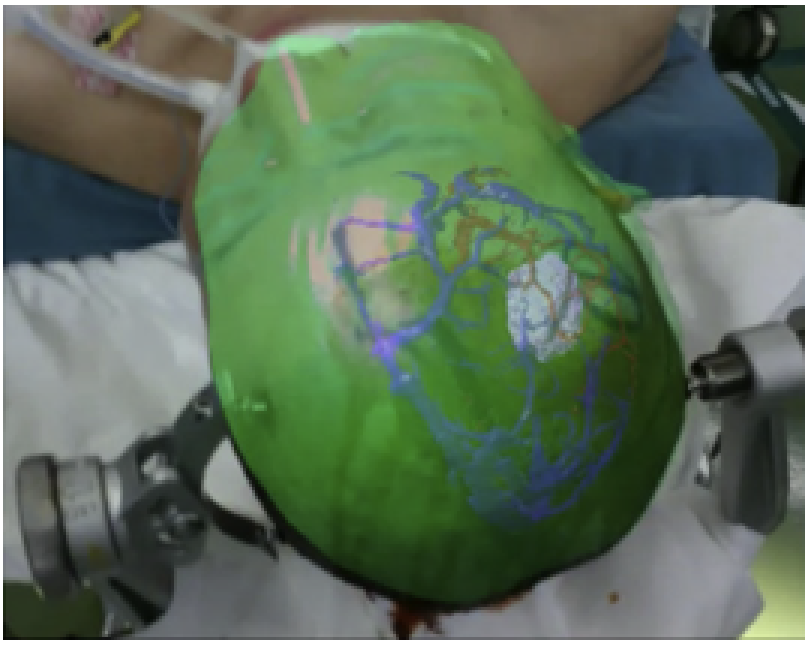
hand-eye coordination from the surgeon, and thus simpler solutions are needed. Inoue et al. (2013) posit that monitor-based systems require that the surgeon look away from the surgical field to see the navigation display, and that a system that can be used without movement of the surgeon's eye would be a vast improvement. Integrating new types of imaging would help solving most of the mentioned neuronavigator shortcomings, but Drouin et al. (2017) report that common neuronavigation systems are not built to accommodate new types of imaging and devices.

Besides the usability limitations, other authors claim that affordable solutions are needed, because neuronavigation systems may have a prohibitive cost, especially in developing countries (LOVO et al., 2007; DWARAKANATH et al., 2007; HOU et al., 2016; EFTEKHAR, 2016; CHEN et al., 2017).

1.3 Augmented reality in neurosurgeries

Augmented reality (AR) has been proposed to address some limitations of the current neuronavigation systems (KERSTEN-OERTEL et al., 2015a; MEOLA et al., 2017; CUTOLO et al., 2017; LÉGER et al., 2017). AR is a technology which allows the superimposition of computer-generated virtual images onto the real world environment (AZUMA, 1997). Thus, AR supplements reality, merging the virtual with the real world. In neurosurgery, it is possible to overlay virtual images, obtained from CT/MRI scans, onto the patient's head (Figure 4). Differently from traditional monitor-based neuronavigation, in which the virtual and the patient spaces are distinct (Figure 2), AR solutions combine those spaces, as shown in Figure 4.

Figure 4 – Virtual image superimposed onto the patient's head, merging the virtual and the patient spaces.



Source – Watanabe et al. (2016)

A study with 12 subjects conducted by Léger et al. (2017) compared three different systems: traditional neuronavigator, desktop AR, and mobile AR. Quantitatively, they analyzed the attention shifts, and the time to delineate a tumor. Regarding attention shifts, they show that the mean ratio of time spent looking at the screen over time taken during tumour delineation was 0.60 ± 0.18 for traditional navigation, 0.91 ± 0.07 for desktop AR, and 0.95 ± 0.05 for mobile AR. The mean times to delineate a tumor were 50.78 ± 24.34s for traditional navigation, 25.5 ± 10.95s for desktop AR, and 20.6 ± 8.23s for mobile AR. Qualitatively, they

used questionnaires to analyze how the subjects felt regarding the systems accuracy, intuitiveness, and comfort. Only one subject (8%) found the traditional navigation more accurate, while 67% found the mobile AR more accurate. All the subjects found the AR solutions more intuitive and comfortable than traditional navigation, of those 83% and 92% preferred mobile AR in terms of intuitiveness and comfort, respectively.

It is worth mentioning that commercial neuronavigators may provide AR capabilities through the use of the surgical microscope, but this increases the cost of the system. Also, microscope-based AR systems are not suitable for the initial part of the surgery (LÉGER et al., 2017; MEOLA et al., 2017), which includes the craniotomy planning. Deng et al. (2014) add that using microscope-based AR in neuronavigation is cumbersome, and, for this reason, it is not widely used in routine neuronavigation.

A variety of display methods have been employed in AR neuronavigation, e.g. tablet (CHANG et al., 2012; DENG et al., 2014; WATANABE et al., 2016), smartphone (EFTEKHAR, 2016; HOU et al., 2016; CHEN et al., 2017; LÉGER et al., 2017), head-mounted display (HMD) (AZIMI; DOSWELL; KAZANZIDES, 2012; ABHARI et al., 2015; CUTOLO et al., 2017; MARUYAMA et al., 2018), video projector (GAVAGHAN et al., 2012; MAHVASH; TABRIZI, 2013; TABRIZI; MAHVASH, 2015), AR window (WESARG et al., 2004; LIAO et al., 2010; FRITZ et al., 2014), and computer monitor (KAWAMATA et al., 2002; GILDENBERG; LABUZ, 2006; KOCKRO et al., 2009; LOW et al., 2010; INOUE et al., 2013; KERSTEN-OERTEL et al., 2015a; GERARD et al., 2018).

Despite the advantages AR systems introduce, they also have some limitations, which may come from two sources: the AR technology itself, and/or the specific AR solution.

Regarding the AR technology limitations, there are three common challenges that most systems have to face: occlusion, depth perception, and inattentional blindness. *Occlusion* is the capability of hiding virtual objects behind real objects. *Depth perception* is how well the user can perceive the distances of the virtual objects. *Inattentional blindness* is the inability to notice an object because the attention is focused on another object or task. These challenges are further discussed in **System requirements** (chapter 4).

Regarding specific AR solutions, we classified them into two groups: *Substitute* and *Affordable*. The former represents the solutions that were proposed as substitutes to neuronavigation systems, e.g. (INOUE et al., 2013; DROUIN et al., 2017), while the latter is composed by solutions that are affordable alternatives when a neuronavigator is not available, e.g. (HOU et al., 2016; EFTEKHAR, 2016).

The *Substitute* systems were developed with the specific purpose of providing better usability than the traditional neuronavigator. While this goal was achieved in many studies, these solutions are also expensive and/or hard to adopt, due to the software and hardware requirements. For instance, the system proposed by Watanabe et al. (2016) uses six cameras, a tablet, a computer, tracking markers, and a complex software that combines the available information to provide the augmented reality experience. The complexity of integrating the hardware and software required to reproduce the *Substitute* approaches may be a barrier for other research groups to explore these solutions because the hardware may be expensive and the software is not easily available.

The *Affordable* solutions are not intended to substitute the neuronavigator because they do not provide neuronavigation. They were proposed to be a low cost alternative to assist in craniotomy planning. Even with this limited applicability, they are still useful because, according to Stadie et al. (2011), several publications pointed out that one of the most valuable applications of frameless neuronavigation was the localization of the craniotomy. Furthermore, in a study presented by Wagner et al. (2000), it was shown that in 40% of the cases where neuronavigation was used, it was only for craniotomy planning. Also, Spivak & Pirouzmand (2005) state

that, in their experience, image-guided systems have been used mostly for craniotomy planning. The limitation of the *Affordable* solutions is that they usually require too many preoperative steps, and thus it does not make sense to use them when a neuronavigator is available. A deeper discussion about the advantages and limitations of recent AR solutions is presented in **Literature review** (chapter 2).

1.4 Problem

Substitute and *Affordable* solutions may be used for craniotomy planning, but they have distinct goals, and thus are applicable in different scenarios. Systems of the *Substitute* group may not be an option in hospitals with few resources, while systems of the *Affordable* group may not be considered in well-equipped hospitals. There is a gap between these groups and a solution that combines the best characteristics of both worlds is needed.

Léger et al. (2017) have shown that mobile devices had better usability than traditional neuronavigators in their tests. In spite of that, most AR neuronavigation systems that use mobile devices as their display method need the device to be connected to an external computer to process the scene information and provide the augmented reality experience. In other words, these approaches need more hardware, are network dependent, and do not use most of the device processing power. It would be useful to have an inexpensive mobile approach that is simple to use and also does not need additional hardware.

Given the current state-of-the-art of AR neuronavigation systems and the findings of Léger et al. (2017), it is needed to have a standalone mobile AR neuronavigator that combines the navigation capabilities of the *Substitute* group and the costs of the *Affordable* group. In order to achieve this, the mobile application has the following requirements: (a) use the device camera and other sensors to locate the patient's head in the physical space and represent it in a form that can be used by the algorithms; (b) perform a registration to map the CT/MRI to the location of the patient's head; (c) render the CT/MRI superimposed on the patient's head. Each of these requirements also brings some challenges. In order to achieve a good accuracy, requirement (a) demands good sensor calibration, precise synchronization between the sensors, and robust implementations of the Computer Vision (CV) algorithms; requirement (b) needs an appropriate method to perform the map from the CT/MRI coordinate system to the patient's head coordinate system; and requirement (c) demands that the CT/MRI is loaded into the mobile application, and then properly rendered to superimpose the patient's head and clearly indicate the incision region. In this work, due to time restrictions, we simplified requirement (c), so that only the center of the incision region should be indicated by the application, without the need of complete CT/MRI superimposition.

1.5 Hypotheses and objective

Modern high-end mobile devices have powerful central processing units (CPUs), graphics processing units (GPUs), and sensors, e.g. accelerometer, gyroscope, and camera. For this reason, we posed two hypotheses:

- It is possible to use Visual Inertial Odometry (VIO) and Computer Vision (CV) techniques to map the environment, and accurately indicate the craniotomy location, superimposed on the patient's head through augmented reality, using only a mobile device, i.e. no external hardware is needed to perform the computations.

- It is possible to achieve significantly shorter registrations times than the ones of the *Affordable* group by using image recognition techniques to identify the registration points instead of relying on manual adjustments.
- It is possible to develop a prototype application to test our hypotheses in the time frame of 1 year using an AR framework to speed up the development time.

In order to test our hypotheses, the objective of this work was to develop a mobile application that: (a) can accurately indicate the incision region on the patient's head, using augmented reality, without the need of additional hardware, and (b) requires less registration time than solutions of the *Affordable* group.

1.6 The application

We introduce a mobile AR application named ARNeuro. This app is similar to the *Affordable* solutions, but has the advantage of not requiring many preoperative steps. In fact, it takes no more than 3 minutes to pass through all the steps of the application, which are: (a) inform the coordinates of 3 registration points and 1 target point; (b) mark the registration points on the patient's head with special image markers; (c) let the application identify the image markers to perform registration; (d) check the target point drawn by the application, superimposed onto the patient's head. A registration point can be any point easily identifiable by the surgeon in the virtual image and on the patient's head. The target point indicates the center of where the incision is going to be performed. These coordinates are easily obtained from the CT/MRI of the patient and a medical imaging viewer software.

ARNeuro provides an accurate, affordable, easy, and fast way of finding the location of the craniotomy for minimally invasive surgeries. Also, this application can be easily distributed, which facilitates future research. The current version is not supposed to perform neuronavigation or to substitute a neuronavigator, because it does not superimpose a CT/MRI on the patient's head. However, implementing this feature is just a development effort, since the application already has the required resources.

Contrary to the *Affordable* solutions, our application may be used when a neuronavigator is available, because its simple usability does not demand too much time from the surgeon. In addition, this approach can also substitute the *Affordable* solutions when a neuronavigator is not available.

The current limitation of our method is that, since it relies on the device sensors, the accuracy may vary depending on some factors, like device handling, electromagnetism interference, etc. For example, exaggerated device motions may negatively impact the accuracy. To handle situations where the accuracy is compromised, the application informs the user that a registration needs to be performed again. A new registration usually takes less than 30 seconds.

The system accuracy was evaluated through the calculation of the target registration error (TRE), a common metric used in literature (WILES et al., 2008). The TRE was defined as the distance between the *virtual target* and the *real target*, which are corresponding points shown on the device screen. The virtual target is the incision point inferred by the application. The real target is the correct incision point. Furthermore, the registration time was also evaluated, since this metric is commonly cited in publications about AR neuronavigation (GUHA et al., 2017). In addition, the alignment between the image registration points identified by the application and their respective registration points on the virtual image was analyzed through

the calculation of the Root-Mean-Square Error (RMSE) of the distances between the corresponding points. This is similar to fiducial registration error (FRE), but it is not the same. More details about the methods are given in **Material and methods** (chapter 5).

Two experiments were performed to evaluate the system. In the first, we used a styrofoam sphere to emulate the human cranium, and analyzed the system accuracy and registration time. In the second, we used a phantom head with its respective CT image, and measured the system accuracy, registration time, and registration points RMSE of two different iPhone models.

In the first experiment, the mean TRE ($\pm$ standard deviation) was 4.3 ($\pm$ 2.4), for the localization of a point in an emulated temporal lobe, and 3.0 ($\pm$ 1.7), for the localization of a point in an emulated frontal lobe. In the second experiment, the mean TRE was 2.989 ($\pm$ 1.607) with the iPhone 6S Plus, and 3.102 ($\pm$ 1.566) with the iPhone 7 Plus. The detailed results are shown in **Results and discussion** (chapter 6).

While more experiments are still needed, especially in clinical cases in the OR, ARNeuro seems to be a promising tool. The current accuracy is still not better than that of other AR systems or traditional neuronavigators, but there are many improvements that can be done to our current approach. Furthermore, our results suggest that the system is better than traditional techniques for craniotomy localization (SPIVAK; PIROUZMAND, 2005).

1.7 Document structure

The rest of this document is organized as follows:

Chapter 2 - Literature review: A systematic review was done in order to select relevant papers of the last 5 years. Then, we present quantitative and qualitative analyses of the AR systems found in literature. Finally, the main advantages and limitations of the solutions are discussed.

Chapter 3 - Theory: Background of the concepts and techniques used in this work explained in a high-level manner.

Chapter 4 - System requirements: The problem is break down into smaller subproblems to provide a better overview of what has to be solved. These subproblems are presented as functional and non-functional requirements.

Chapter 5 - Material and methods: Description of what was used to develop and test ARNeuro. Also, it is detailed how the user should use the application for craniotomy planning. Finally, implementations details are discussed.

Chapter 6 - Results and discussion: Results of two different experiments are shown and discussed. We analyze the system accuracy, registration time, and RMSE of the matching between the registration points.

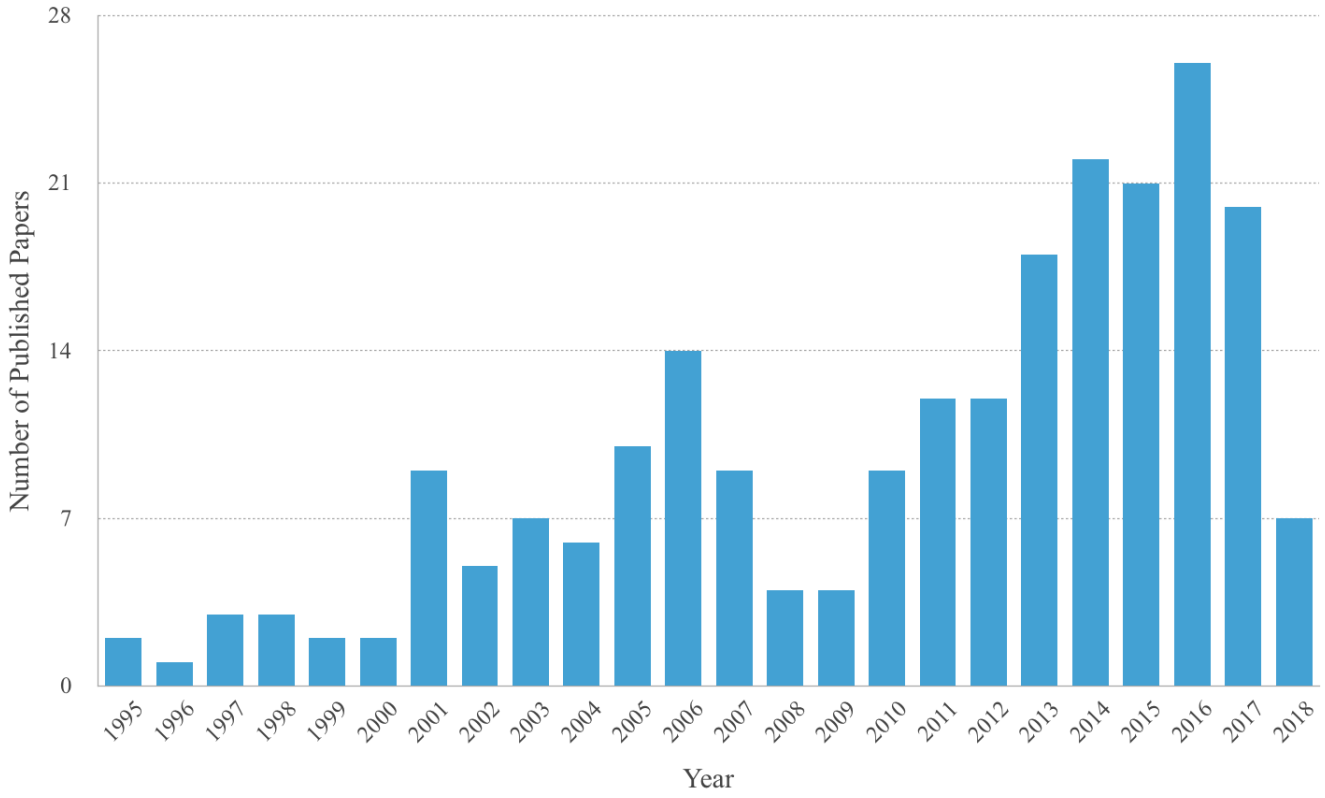
Chapter 7 - Conclusion and future work: The findings of this work is summarized and we present some of the possibilities for future work.

2

Literature review

Neurosurgical applications of AR began in the mid-1980s, when Roberts et al. (1986) presented the first augmented operating microscope (GUHA et al., 2017). In the last years, the use of AR in neurosurgery has shown a tendency of growing popularity, as it can be seen by the number of published papers by year (Figure 5).

Figure 5 – AR in neurosurgery published papers by year.



Source – Non-duplicated results from the search explained in Section 2.2.

Currently, there is no single best solution. Every AR method has its pros and cons. For example, microscope-based AR systems, which combine AR technology with surgical microscopes, are not ergonomically practical for the initial macroscopic part[1] of the surgical procedure (MEOLA et al., 2017; LÉGER et al., 2017). Furthermore, alternatives presented in literature have minimal scope (TAGAYTAYAN; KELEMEN; SIK-

[1] Skin incision, craniotomy, and dural opening.

LANYI, 2016; MEOLA et al., 2017; GUHA et al., 2017), meaning that each solution applies to very specific neurosurgical situations, like training (ALARAJ et al., 2013; ABHARI et al., 2015), neurovascular surgery (KERSTEN-OERTEL et al., 2015b; CABRILO; SCHALLER; BIJLENGA, 2015), tumor resection (KERSTEN-OERTEL et al., 2016; WATANABE et al., 2016), spine surgery (ABE et al., 2013; WU et al., 2014), and so on.

Given the growing research field and the heterogeneous published work, a systematic review was elaborated with the goals of identifying solutions that solve a similar problem to ours, and analyzing their results to have a comparative reference for our approach.

All the steps of this systematic review were performed by this author only, but, in order to provide better results, three recent literature reviews (TAGAYTAYAN; KELEMEN; SIK-LANYI, 2016; MEOLA et al., 2017; GUHA et al., 2017) were used to complement and to double-check our findings. The following sections detail the review.

2.1 Question

The question that guided this review was: *How the state-of-the-art AR methods that assist in craniotomy planning for tumor resection compare to each other?*

The idea behind this question was to find relevant work that could be compared, quantitatively and qualitatively, with our proposed approach.

2.2 Search

A preliminary search was done in July, 2017 and an updated revision in May, 2018. We aimed at a broad-scope search in order to avoid missing important work. The general search string was "*(augmented reality) AND (neurosurgery OR neurosurgical)*". The search fields were usually restricted to *title*, *abstract* and *keywords*. There was no restriction on the *date* parameter, but the oldest result was from 1995. The search engines were PubMed, Scopus, IEEE Xplore, Web of Science, Science Direct, and Engineering Village. Details of the search are exposed on Table 1.

Table 1 – Systematic review search details.

Search Engine	**Search String**	**Search Fields**
PubMed	((augmented reality[Title/Abstract]) AND (neurosurgery[Title/Abstract] OR neurosurgical[Title/Abstract]))	Title, Abstract
Scopus	"augmented reality" AND (neurosurgery OR neurosurgical)	Title, Abstract, Keywords
IEEE Xplore	augmented reality AND (neurosurgery OR neurosurgical)	Metadata (title, abstract, keywords)
Web of Science	TS=(augmented reality) AND TS=(neurosurgery OR neurosurgical)	Topic (title, abstract, keywords)
Science Direct	(augmented reality) AND (neurosurgery OR neurosurgical)	Title, Abstract, Keywords
Engineering Village	augmented reality AND (neurosurgery OR neurosurgical)	Title, Abstract, Subject

The search retrieved 491 results.

2.3 Selection

The software StArt (ZAMBONI et al., 2010), version 2.3.4.2, was used to remove duplicated papers and select the ones that would be relevant for deeper analysis. Before removing duplicates, though, the three literature reviews previously mentioned were scrutinized. All the papers that could not be found in our search but were discussed in any of these previous works were added to our review. 11 papers were added this way, as shown in Table 2.

Table 2 – Papers not found in the search phase detailed in section 2.2 but identified in other literature reviews.

Author(s) and year	Title
Abe et al. (2013)	A novel 3D guidance system using augmented reality for percutaneous vertebroplasty
Wu et al. (2014)	Real-time advanced spinal surgery via visible patient model and augmented reality system
Das et al. (2006)	Augmented reality visualization for CT-guided interventions: system description, feasibility, and initial evaluation in an abdominal phantom
Wacker et al. (2006)	An augmented reality system for MR image–guided needle biopsy: initial results in a swine model
Kantelhardt et al. (2015)	Video-assisted navigation for adjustment of image-guidance accuracy to slight brain shift
Gildenberg & Labuz (2006)	Use of a volumetric target for image-guided surgery
Bisson, Cheriet & Parent (2010)	3D visualization tool for minimally invasive discectomy assistance
Navab, Heining & Traub (2010)	Camera augmented mobile C-arm (CAMC): calibration, accuracy study, and clinical applications
Shenai et al. (2011)	Virtual interactive presence and augmented reality (VIPAR) for remote surgical assistance
Weiss et al. (2011)	Augmented reality visualization using image-overlay for MR-guided interventions: system description, feasibility, and initial evaluation in a spine phantom
Fritz et al. (2014)	MR-guided vertebroplasty with augmented reality image overlay navigation

After adding the papers from Table 2 and removing duplicates, there was a total of 226 unique papers. These were submitted to an abstract screening process using the exclusion and inclusion criteria explained in sections 2.3.1 and 2.3.2, respectively.

2.3.1 Exclusion criteria

- Images displayed on external monitor

  As pointed out by Inoue et al. (2013), a system which does not require the surgeon to look away from the surgical field is a great improvement. ARNeuro was designed to allow in-situ visualization, which is not possible with external monitors. Since our method was implemented for mobile devices, and external monitors provide a really different usability, we decided that it would be better to compare our system with others that have similar usability.

- Research published with more than 5 years

  The search parameters did not include any date restriction because some old work could be relevant. After some initial investigation, though, it was observed that old AR systems were limited, mostly because of the available hardware at that time. The majority of older work was already excluded because their systems used external computer monitors. Furthermore, since this area is relatively new and there are not many published papers, recent work already cite many previous approaches, making possible for us to decide if any of them should be considered as an exception and included for comparison. Lastly, we were trying to identify and analyze only the state-of-the-art solutions, because they usually address issues that previous systems had.

- Training/Simulation

  It was decided to exclude training and simulation applications from this review because our main concern is in improving the real neurosurgery procedure. This work also does not have proper metrics to validate the current approach in a learning environment. Since the proposed application can be used for simulation or training, this may be considered for future work.

- Endoscopic, vascular or spine surgery

  The current solution is focused on finding the incision region for a standard craniotomy procedure in a tumor resection surgery. This means that the current state of the application is not suitable for other kinds of surgeries and thus it cannot be properly compared with systems developed for endoscopic, neurovascular or spine surgeries.

- Robotic neurosurgery

  The developed application has no use in robotics in its current state. Since one of our goals is to evaluate the system usability by the human doctor, there was no need to include research focused on robotic systems.

- No experiments or limited results

  Many studies only advocated about the use of AR, without showing any experimentation report. Other studies did not present clear results or metrics. Those cannot be used for comparison. It is worth noting that literature reviews do fit in this category, since they do not present original experimentation results.

- AR in other areas

  Some search results were related to non-neurosurgery areas, e.g. dentistry, so they are meaningless in this context.

- Not AR

  Work that do not use AR cannot be directly compared with our approach because many metrics are very different.

- Specific topic

  Topics such as *camera calibration* and *neurosurgery workflow* are too specific and do not offer much value for our research because the papers lack a complete system analysis for comparison.

- No access

  A small number of papers were discarded because we did not have access to them.

2.3.2 Inclusion criteria

- AR applied for standard craniotomy in tumor resection surgery

  The goal of our application is to assist the surgeon in finding the incision region to perform a craniotomy procedure in a tumor resection surgery. For this reason, every AR application that fits the same role should be considered for analysis, even if they use different devices.

- Mobile devices

  The current developed AR system works on smartphones and tablets. We concluded that alternatives using similar devices should also be considered for analysis because they share an important part of our method. The use of mobile devices is one of the things that improves the system usability when compared to other solutions that require more complicated setup.

- Relevant old papers

  Some papers cite relevant work with more than 5 years of published. So, even if some of these works may be outdated compared to the state-of-the-art, they are considered worth including in our selection.

2.3.3 Selection results

Applying the selection criteria resulted in a total of 11 relevant papers.

The only work that fits an exclusion criterion, i.e. *Research published with more than 5 years*, but was included in the results, is Chang et al. (2012)'s, because, despite being from 2012, it shares some similarities with our method, and it also fits all the inclusion criteria.

Some other papers matched one or more inclusion criteria, but they also matched more relevant exclusion criteria, and therefore they were not included.

2.4 Analysis

In previous systematic reviews (GUHA et al., 2017; MEOLA et al., 2017), the analysis of the studies were presented with different focus. Guha et al. (2017) provide a better quantitative analysis, because their review discuss the reported outcomes, e.g. registration error, of the selected studies. Meola et al. (2017), on the other hand, provide a better qualitative analysis, because they give more emphasis on technical implementation, advantages, and limitations of each solution.

In this review, the selected studies are evaluated quantitatively and qualitatively. Guha et al. (2017)'s parameters were chosen for quantitative analysis (section 2.4.1), providing a general idea of the systems accuracy. Meola et al. (2017)'s parameters were chosen for qualitative analysis (section 2.4.2), providing more details about the technical approaches.

2.4.1 Quantitative analysis

According to Guha et al. (2017), there are two common quantified metrics adopted by most clinical studies: *setup time* and *overall registration error*. In this work, it was decided to use *registration time* instead of *setup time*, because some papers consider different phases in the calculation of the latter, making it less viable for comparison between systems. The advantage of the chosen metrics is that both can be used to evaluate and compare different AR systems.

Guha et al. (2017) also point that the composition of the overall registration error differs between the studies. The reason for this is that different techniques introduce different errors. For instance, some systems may have a tracking error (DENG et al., 2014; WATANABE et al., 2016), while others do not have this problem (TABRIZI; MAHVASH, 2015; HOU et al., 2016). Nevertheless, the overall registration error is still a useful

metric because, not only it serves as a common ground to compare two different solutions, but it also indicates if a technique can be used in a specific case.

One thing that should be emphasized is that even though the quantitative metrics provide valuable information they are not the only relevant factors when evaluating an AR system. In some situations, for example, the surgeon may opt for a slightly less accurate system, that has better usability, instead of one that is more accurate but requires difficult setup. For this reason, the results exposed in this subsection should be analyzed in conjunction with the qualitative analysis of section 2.4.2.

The summary of the quantitative analysis is presented in Table 3.

Table 3 – Results of the quantitative analysis of the selected AR systems.

Author(s) and year	**No. of cases**	**Display device**	**Registration error***	**Registration time***
Chang et al. (2012)	1 phantom	Tablet	2.2mm	—
Mahvash & Tabrizi (2013)	1 phantom	Video projector	0.3mm (range: 0.1-0.6 mm)	5 minutes
Deng et al. (2014)	1 cadaver skull 2 patients	Tablet	Cadaver skull: 1.6mm Clinical: 2.1mm	—
Tabrizi & Mahvash (2015)	10 phantoms 5 patients	Video projector	Phantom: 0.8 $\pm$ 0.25mm Clinical: 1.2 $\pm$ 0.54mm	Phantom: 3.8 minutes
Eftekhar (2016)	11 patients	Smartphone	10.2 $\pm$ 2.0mm	4 $\pm$ 1 minutes
Watanabe et al. (2016)	1 phantom 6 patients	Tablet	Phantom: ~1.0mm	Clinical: 3 minutes
Hou et al. (2016)	35 patients	Smartphone	~5.0mm	10 minutes
Chen et al. (2017)	16 patients	Smartphone	4.4 $\pm$ 1.1mm	141.7 $\pm$ 39 seconds
Léger et al. (2017)	1 phantom	Smartphone	1.76mm	—
Cutolo et al. (2017)	1 phantom	HMD	—	—
Maruyama et al. (2018)	1 phantom 2 patients	HMD	Phantom: 3.1 $\pm$ 1.9mm Clinical: 2.1 $\pm$ 1.1mm	3 minutes

* All values are presented as means.

In order to achieve the best possible comparison, it was decided to use *registration time* and *overall registration error* as the quantified metrics, for the reasons previously mentioned in this subsection. The *number of cases* in which the system was tested was taken in consideration because it indicates how trustworthy the error estimation is. The *display device* is mentioned to show that every device can achieve good accuracy, because the registration error is only affected by the registration method. A deeper discussion about the display devices and the registration techniques is left for the qualitative analysis (section 2.4.2).

Some of the papers present the same techniques, with changes in the number of test cases and/or in their methodologies. The video projector technique, initially introduced by Mahvash & Tabrizi (2013), was further tested by Tabrizi & Mahvash (2015), which present a study with more tests and thus better error assessment. The works presented by Watanabe et al. (2016) and by Maruyama et al. (2018) use different display devices, but a similar system for registration and tracking. There is a difference in the number of cameras - the previous approach uses 6, while the recent uses *at least* 2. Their registration errors differ a little and this may be due to the improvement in the test methodology and also to the number of cameras. The most interesting case is the smartphone technique used by Eftekhar (2016) and by Chen et al. (2017). Both use the same application, but the registration technique applied in the more recent work was changed to include fiducial markers, which greatly reduced the registration error.

Considering the mentioned facts, we may assume that the results by Mahvash & Tabrizi (2013) and by Eftekhar (2016) are outliers. The former had its technique better tested in a more recent work (TABRIZI; MAHVASH, 2015), and the latter's technique had improved results after the methodology change shown in

(CHEN et al., 2017). Most of the other techniques have registration errors between 1.0mm and 3.1mm, which is similar to the accuracy achieved by current neuronavigation systems (GUHA et al., 2017; MASCOTT et al., 2006; WOERDEMAN et al., 2007). In spite of that, with the exception of the study presented by Hou et al. (2016), all the works had less than 30 cases for testing, even though most of them made many measurements of the same cases to have more accurate data. Cutolo et al. (2017) did not present quantified metrics in their study but the paper was included in this review because it brings useful qualitative metrics.

2.4.2 Qualitative analysis

High accuracy is desired in any system, especially when it involves a complicated surgical procedure. However, after achieving the minimal required accuracy, other important attributes become important, like system usability and feasibility. In this subsection we evaluate the selected solutions in a qualitative manner. For this task, we also took the opinion of a specialist, Bruno Fernandes de Oliveira Santos, MD, MSc.

According to Meola et al. (2017), there are ten parameters that should be considered to analyze AR systems in neurosurgery. These parameters and some of their possible values are shown in Figure 6.

Figure 6 – Meola et al. (2017)'s suggested parameters to analyze an AR system in neurosurgery. *Field of use* column represents the parameters and *Options (examples)* column represents the possible values for the parameters.

Field of use	Options (examples)
1. Open neurosurgery	Macroscopic (allowed by all systems) Microscopic (only microscope and endoscope)
2. Endoscopy	Endoscope
3. Endovascular neurosurgery	X-ray fluoroscopy
AR system features	
4. Real data source	Microscope video camera (hand-held or head-held) Tablet-camera unaided view (projector; light field display and half mirror at 45°, LFD) Endoscope X-ray fluoroscopy
5. Tracking modality	Optical (broadly used for all the systems) Magnetic (used only by Doyle WK, 1996) None
6. Registration technique	Fiducial markers (applied to skin, teeth, skull) Skin surface registration Manual
7. Display type	Microscope oculars semitransparent head-up display (used only by Doyle WK, 1996 and questionable) External monitor (hand video camera, camera, microscope, endoscope, X-ray fluoroscopy) Tablet monitor LFD Projector
8. Perception location	Patient (microscope, tablet, projector, LFD) External monitor
AR scene parameters	
9. Virtual image source	CT MRI (and related techniques) Angiography
10. Visualization	Wireframe texture map surface mesh transparencies light field rendering standard navigation view

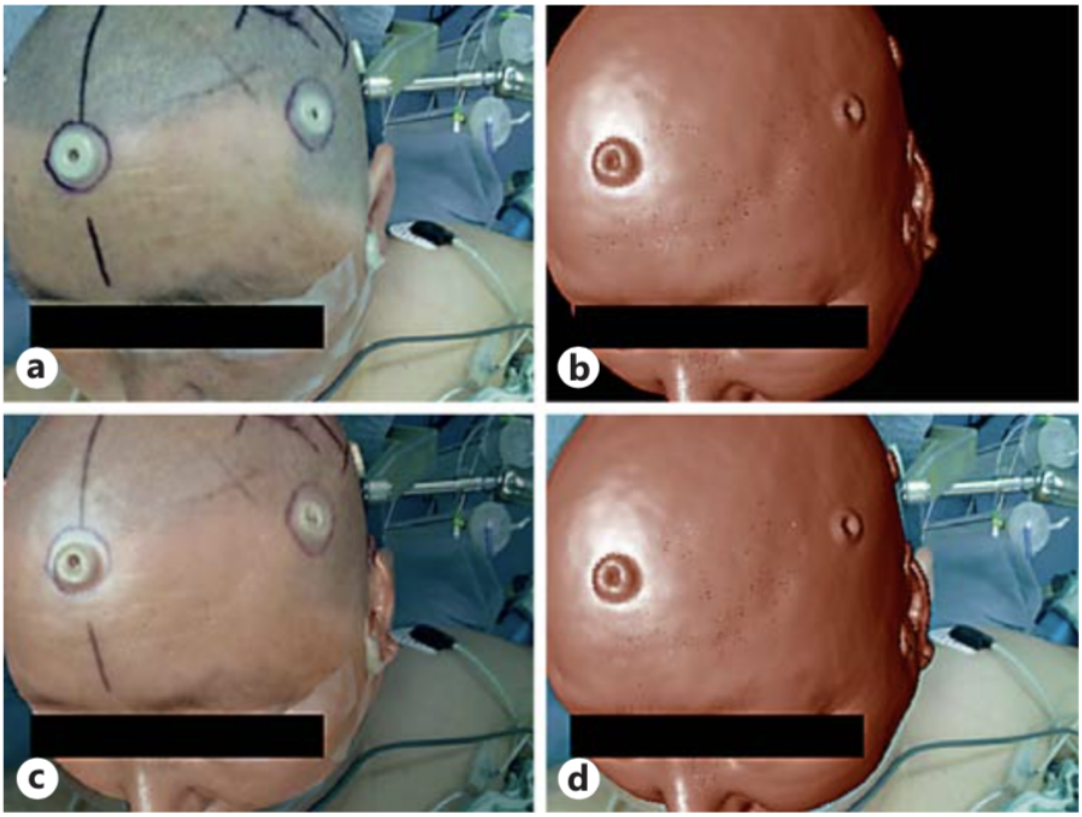
Source – Meola et al. (2017)

Parameters 1 to 3 indicate the type of surgery. *Open neurosurgery* is a category in which a craniotomy has to be performed, to open the patient's head, while *Endoscopy* and *Endovascular neurosurgery* do not involve a craniotomy procedure.

Parameters 4 to 8 indicate the AR system features. *Real data source* represents the device that captures the images from the world, e.g. camera. The images provided by the real data source are the ones that will be superimposed by the virtual images. *Tracking modality* is how the AR system performs tracking. *Registration technique* indicates what is used to perform the registration of the patient space in the AR system. *Display type* is the device which displays the images obtained from the real data source superimposed by the virtual images. *Perception location* is where the surgeon perceive the superimposition of the virtual image. This can be on an external monitor, or on the patient, i.e. in situ, through the use of a display device.

Parameters 9 and 10 are related to the AR virtual environment. *Virtual image source* indicates how the virtual image is obtained. *Visualization* indicates how the virtual image is overlaid. A surface mesh, for example, is a visualization in which the virtual image meshes with the surface of the patient's head (Figure 7).

Figure 7 – Surface mesh visualization. **A**: Surgery scene. **B**: Virtual image. **C**: Superimposition of a 50% transparent virtual image onto the patient's head. **D**: Superimposition of an opaque virtual image onto the patient's head.



Source – Deng et al. (2014)

Given the restricted scope and the purpose of this review (section 2.1), many parameter options were discarded by the selection phase (section 2.3), originating a modified version of Meola et al. (2017)'s table. Table 4 shows only the values that are pertinent to our review in the options column. Parameter 1 has only one possible value, i.e. *Macroscopic*, because, as stated in the beginning of this chapter, microscopic-based AR systems are not suitable for the initial part of the surgical procedure, and, since our review is focused on craniotomy planning, all selected solutions were focused on the macroscopic phase of open neurosurgeries. For the same reason, parameters 2 and 3 have no possible options because those types of surgery were discarded by the exclusion criteria (section 2.3.1). Parameter 8 has also only one possible value because solutions that use *external monitor* were discarded by the exclusion criteria (section 2.3.1). The other parameters (4, 5, 6, 7, and 10) are useful in this analysis and are used to classify the selected solutions.

Table 4 – A version of Meola et al. (2017)'s table, modified to include only parameter options of solutions presented in this review.

Parameter	Options
1. Open neurosurgery	Macroscopic
2. Endoscopy	—
3. Endovascular neurosurgery	—
AR systems features	
4. Real data source	Device camera Unaided
5. Tracking modality	Optical None
6. Registration technique	Fiducial markers Manual
7. Display type	Video projector Tablet Smartphone HMD
8. Perception location	Patient
AR scene parameters	
9. Virtual image source	CT/MRI
10. Visualization	Surface mesh Transparency

As it can be inferred from Table 4, all the selected solutions: (a) were developed for open neurosurgeries, focusing on the macroscopic part; (b) have perceptual location on the patient; (c) use CT or MRI as virtual image source. Furthermore, solutions that perform *optical tracking* also use *fiducial markers registration* technique, and solutions that *do not perform tracking* use a *manual registration* technique. For this reason, the summary of the systems characteristics is shown in two tables. Table 5 shows the solutions that do not perform tracking and use a manual registration technique, which we called the *Affordable* group. Table 6 shows the solutions that perform optical tracking and use fiducial markers registration technique, which we called the *Substitute* group.

Table 5 – *Affordable* group. Solutions with no tracking and manual registration technique.

Author(s) and year	**Real data source**	**Display type**	**Visualization**
Mahvash & Tabrizi (2013)	Unaided	Video projector	Surface mesh
Tabrizi & Mahvash (2015)	Unaided	Video projector	Surface mesh
Eftekhar (2016)	Device camera	Smartphone	Transparency
Hou et al. (2016)	Device camera	Smartphone	Transparency
Chen et al. (2017)	Device camera	Smartphone	Transparency

Table 6 – *Substitute* group. Solutions with optical tracking and fiducial markers registration technique.

Author(s) and year	**Real data source**	**Display type**	**Visualization**
Chang et al. (2012)	Device camera	Tablet	Surface mesh
Deng et al. (2014)	Device camera	Tablet	Surface mesh
Watanabe et al. (2016)	Device camera	Tablet	Surface mesh
Léger et al. (2017)	Device camera	Smartphone	Surface mesh
Cutolo et al. (2017)	External cameras	HMD	Surface mesh
Maruyama et al. (2018)	Unaided	HMD	Surface mesh

From Table 5 and Table 6, it is possible to see that mobile devices, i.e. smartphones and tablets, use the device camera as real data source. Video projector and HMD usually do not have integrated cameras, so they rely on external cameras (CUTOLO et al., 2017) or they do not need to use cameras (MAHVASH; TABRIZI, 2013; TABRIZI; MAHVASH, 2015; MARUYAMA et al., 2018). *Surface mesh* is an intuitive visualization method (MEOLA et al., 2017), and this is probably the reason why most solutions chose to use it. *Tracking modality* and *registration technique* are the main characteristics that impact in the system usability, so they are discussed with more detail in the rest of this subsection.

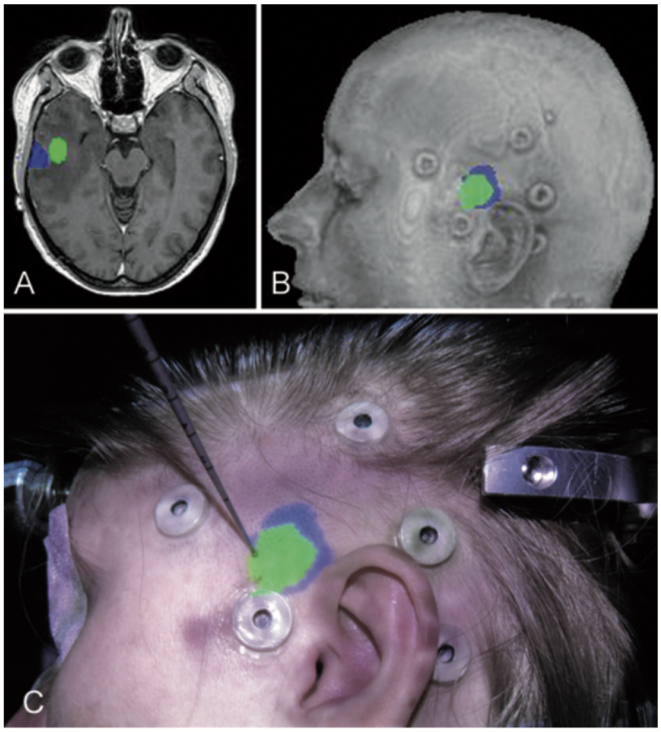
Systems with manual registration and no tracking (*Affordable* group)

The systems presented by the papers in Table 5 perform manual registration and no tracking. With this type of registration, the user has to execute some manual steps in order to align the virtual image, i.e. CT/MRI, with the real data, i.e. patient's head. The absence of the marker tracking capability indicates that these solutions cannot perform neuronavigation. The main advantage of approaches that use this registration is that, besides the display device, no additional equipment is required to perform the image overlay. Consequently, these systems are usually more affordable and easier to adopt than systems with tracking capabilities. The main disadvantage is that the registration may take some time and is usually a tedious task.

In the works of Mahvash & Tabrizi (2013) and Tabrizi & Mahvash (2015), a video projector is used as a display device. The technique consists in projecting a virtual image directly on the patient's head using a common video projector. The user must adjust the projector position and configuration to perform the manual registration (Figure 8). The advantages of this technique are: (a) low registration error; (b) low cost; (c) the neurosurgeon can focus directly on the patient's head. The limitations of this technique are: (a) the projector and its alignment requirements may interfere with the surgeon, the microscope, and the instruments (TABRIZI; MAHVASH, 2015); (b) image distortion may be caused in curved surfaces, i.e. peripheral regions of the head (TABRIZI; MAHVASH, 2015); (c) the different viewpoints of the projector and the surgeon causes a parallax error, which is increased for deep lesions (WATANABE et al., 2016; FERRARI; CUTOLO, 2016; CUTOLO et al., 2017); (d) manual adjustment of focus, size, and position, of a projected virtual image, is a tedious process, especially when a patient is in a surgical position (HOU et al., 2016). This approach might be used in situations where the lesion is located close to the brain surface, but this limited application may not compensate the required setup, i.e. positioning the video projector in the room and performing manual registration.

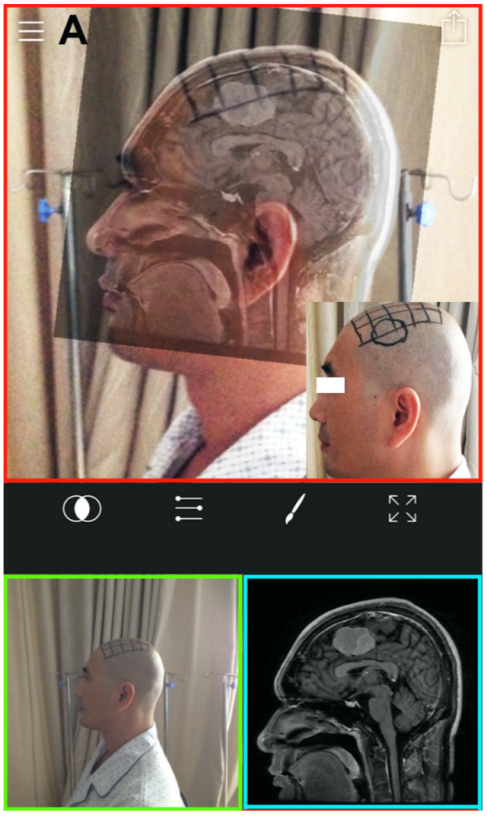
In the works of Eftekhar (2016), Chen et al. (2017), and Hou et al. (2016), smartphones are used as display devices. Both techniques rely on a manual registration step, which consists in aligning a virtual 2D image with the displayed camera feed (EFTEKHAR, 2016; CHEN et al., 2017) or with a sagittal photograph (HOU et al., 2016). The alignment required by Hou et al. (2016)'s technique is shown in Figure 9. Since Chen et al. (2017) present an updated and more accurate approach than Eftekhar (2016), only the first will be considered in this evaluation. The advantages of these smartphone solutions are: (a) acceptable registration errors (HOU et al., 2016); (b) ubiquitous devices; (c) low cost; (d) easy setup. The limitations of these techniques are: (a) not as accurate as a neuronavigator; (b) requires manual registration; (c) do not allow the view from different perspectives due to the 2D superimposition nature of the technique; (d) requires many preoperative steps. Hou et al. (2016)'s technique requires drawing a grid on the shaved head of the patient, acquisition of sagittal image of the patient, preprocessing CT/MRI, transferring the processed CT/MRI to the device, and then perform the registration. Chen et al. (2017)'s technique requires acquiring a CT/MRI with fiducials, transferring the image to the device, and then perform the registration. As the authors state, these techniques are low cost and may be useful in developing countries, when a neuronavigation system is not available.

Figure 8 – Video projection technique presented by Tabrizi & Mahvash (2015). **A**: Preoperative MR image. **B**: MRI-based 3D head model. This image is used for projection onto the patient's head. **C**: Projection of the created image in panel B.



Source – Tabrizi & Mahvash (2015)

Figure 9 – Smartphone technique presented by Hou et al. (2016). The image shows the user interface of the smartphone app adopted in their study. The green box shows the sagittal photograph of the patient, which is superimposed by the MR image shown in the blue box, resulting in the red box image.

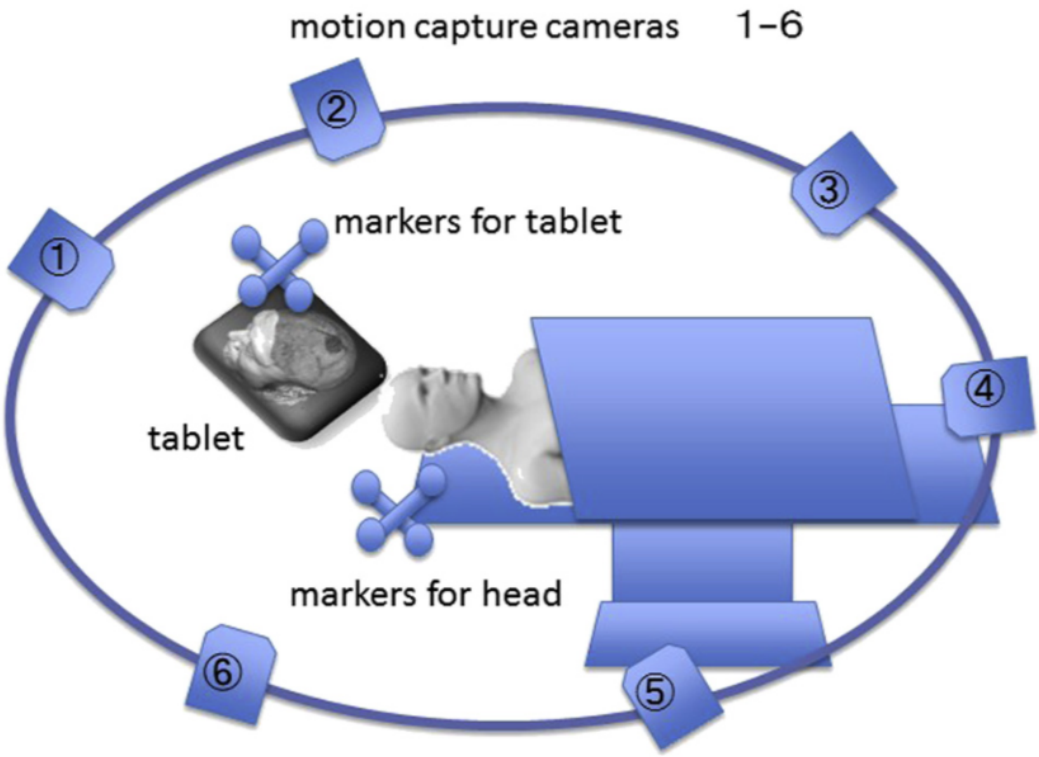


Source – Hou et al. (2016)

Systems with fiducial markers registration and optical tracking (*Substitute* group)

The systems presented by the papers in Table 6 perform fiducial markers registration and optical tracking, which is achieved through the use of reference markers, cameras, and a PC workstation (Figure 10). The idea works as follows: (a) reference markers are attached to the device that should be tracked (HMD, smartphone or tablet) and near the head of the patient; (b) the cameras capture the the images and deliver them to a software installed on the PC workstation; (c) the software process the information from the images, retrieving the locations of the display device and the patient's head; (d) the software performs an image transform on the CT/MRI so that the image can be presented on the display device; (e) the transformed image is sent from the computer to the display device, which shows the image superimposed on the patient's head (Figure 11).

Figure 10 – System diagram presented by Watanabe et al. (2016). Six motion capture cameras are placed on the ceiling of the operation room. Reflective balls are attached to the tablet and to the skull clamp.

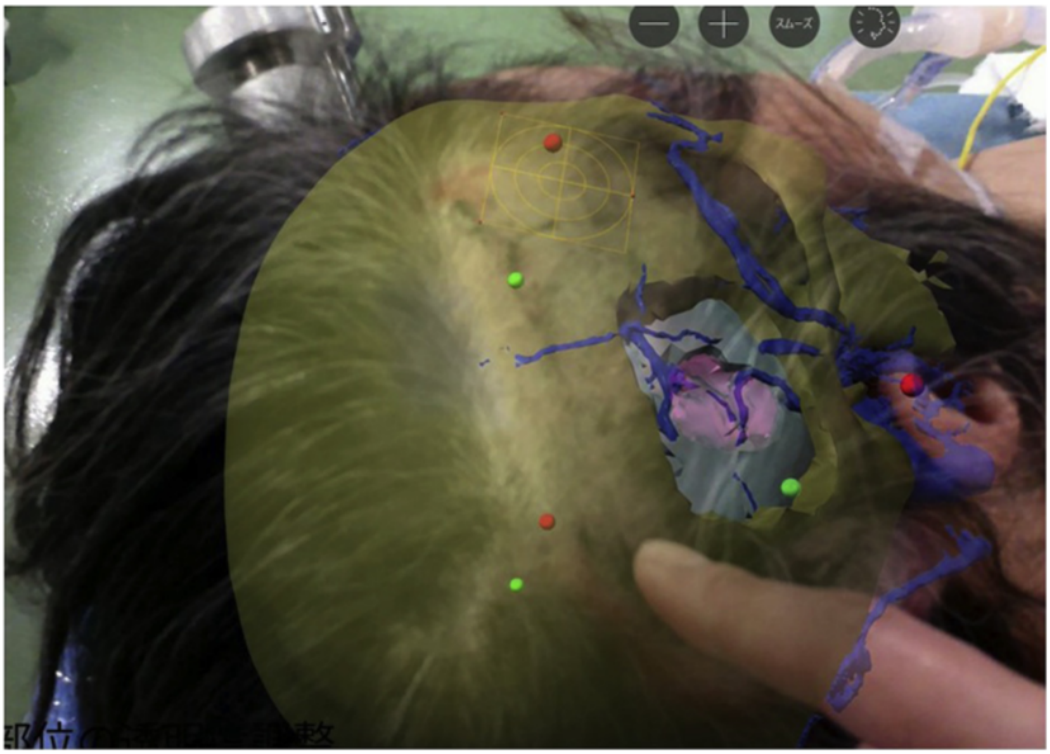


Source – Watanabe et al. (2016)

The general approach is basically the same across the optical tracking solutions, therefore they share some advantages and limitations. The main advantage is that, since the systems perform tracking, it is possible to have a volumetric navigation, instead of just overlaying 2D images on the patient's head. The main limitations are: (a) these solutions require additional hardware for tracking, i.e. cameras, and for processing the images, i.e. PC, which elevates the cost and the complexity of the system; (b) since the approaches rely on the cameras and the markers for tracking, there cannot be something occluding the markers from the camera; (c) even though some of these systems do not require a registration step, in most cases it is necessary to perform a camera calibration before the surgery; (d) when the display device moves, there is a slight delay in updating the virtual image superimposition, due to communication between the display device and the PC. This delay was mentioned as one of the few disadvantages of using a mobile device in Léger et al. (2017)'s study.

Regarding specific display methods, there are two types of HMD devices: (a) video see-through

Figure 11 – Virtual image superimposing the patient's head.



Source – Watanabe et al. (2016)

(Figure 12), which displays the virtual images overlaid onto the camera feed, and (b) optical see-through, which displays the virtual images overlaid onto the user's vision of the world (Figure 13). A video see-through device has the advantages that it minimizes perceptual discrepancies between real and virtual environments by having full control of both (GUHA et al., 2017), and also its view can be displayed on an external monitor. Its limitation is that the display resolution is not good compared to the human eye. On the other hand, an optical see-through device has the advantage of using the surgeon eye's view, but this increases perceptual discrepancies between real and virtual, and also restricts the AR view to the surgeon only.

Both HMD methods give the following advantages: (a) the hands of the surgeon are free to perform the surgery; (b) the usability should feel natural, since the HMD simulates an extension of the eye. The limitations are: (a) wearing an HMD during the surgery can be inconvenient because the device may be too bulky (WATANABE et al., 2016); (b) the HMD needs to be connected to a PC using a cable; (c) the eye position has to be considered in the system setup to avoid unwanted parallax. Cutolo et al. (2017) solves the problem of the eye position by fixing 2 cameras with an anthropometric interaxial distance of ~7cm (Figure 12). Maruyama et al. (2018) solves the same problem by tracking the HMD and registering the eye position (Figure 13).

Tablets and smartphones share the same hardware resources, so any approach that is implemented for one can be easily ported to the other. The main difference between both is the screen size. If a larger screen size is needed, a tablet should be the choice. In this case, the device is usually held by an assistant or by a mechanical arm, since the surgeon cannot hold the tablet and mark the incision at the same time. With a smartphone it is possible to hold the device with one hand and mark the incision with the other. Léger et al. (2017) reported that, in their study, the most common negative comments about the smartphone was the lag in the video feed and the screen size. They attached the device to a mechanical arm (Figure 14).

The main advantage of using smartphones or tablets is that they are easy to use and to integrate in

Figure 12 – Cutolo et al. (2017)'s video see-through HMD device.

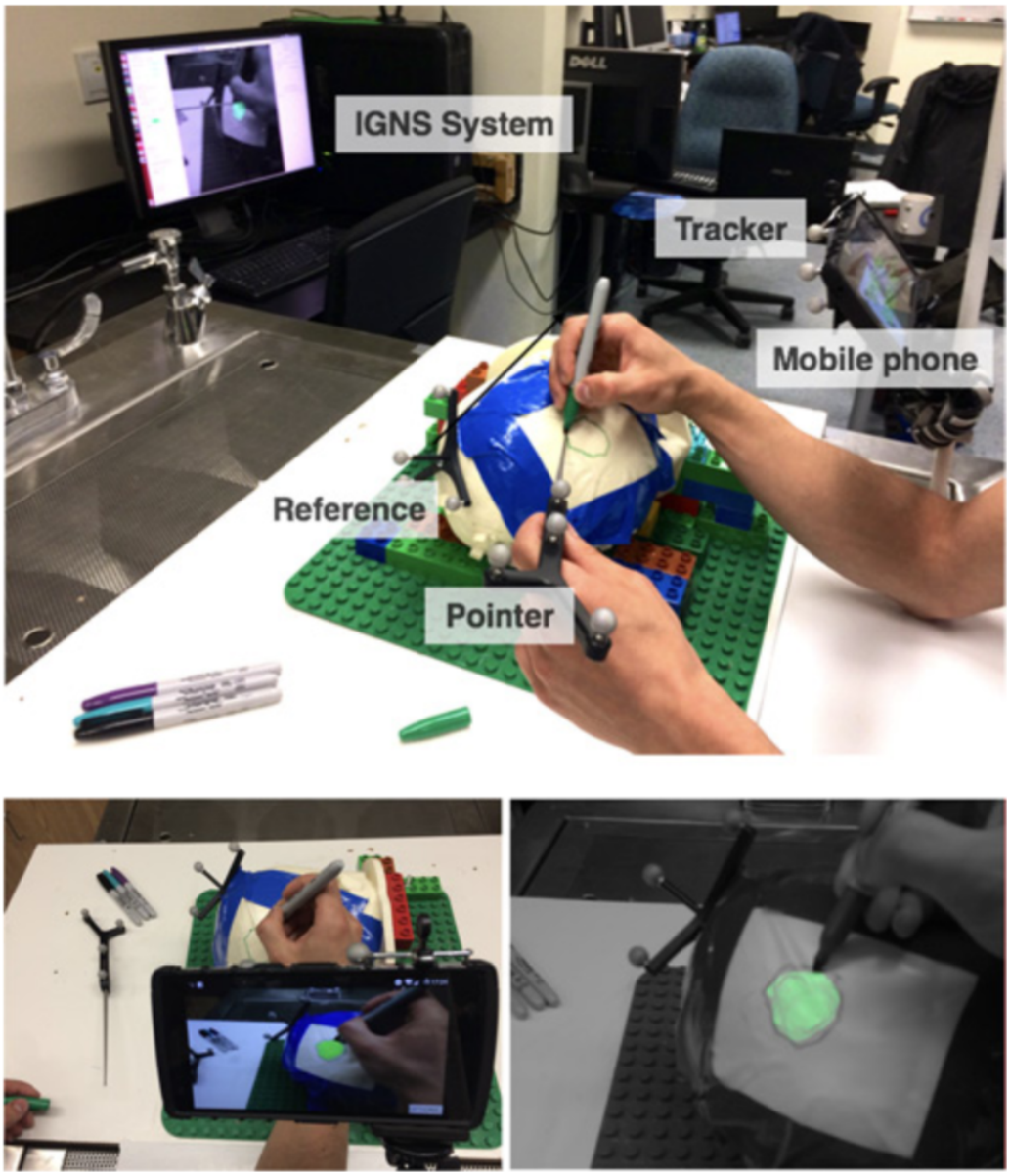
Source – Cutolo et al. (2017)

Figure 13 – Maruyama et al. (2018)'s optical see-through HMD device. The yellow arrows point the markers used for tracking.

Source – Maruyama et al. (2018)

the OR. The main limitations are: (a) the device obstructs a clear working space, because it stays between the surgeon and the patient (EFTEKHAR, 2016); (b) if the device is not physically connected to the PC that process the information, there may be some lag in the video feed, due to the WiFi video transfer from the PC to the device (WATANABE et al., 2016). Compared to the smartphone approaches that use no tracking, approaches that perform optical tracking are more complete, i.e. allow volumetric navigation, and easier to

Figure 14 – Léger et al. (2017)'s smartphone and desktop AR system. Top: complete experiment setup. Bottom-left: surgeon's view, with the smartphone attached to a mechanical arm. Bottom-right: screenshot of the desktop image.



Source – Léger et al. (2017)

setup, assuming that the additional hardware, like the cameras and the PC, are ready for use in the OR. The cost for these benefits is additional hardware required.

2.5 Review conclusion

As it was stated in the beginning of this chapter, currently there is no best AR system. In order to investigate the current solutions, a systematic review was elaborated. First, a broad-scope search was made, with the goal of not missing important research in the area. Then, the papers were submitted to a selection phase with very specific exclusion and inclusion criteria, to select only the most relevant work. Finally, quantitative and qualitative analyses were conducted, the former following Guha et al. (2017)'s approach, and the latter Meola et al. (2017)'s approach.

As it is shown in the analysis section, systems with optical tracking and fiducial markers registration

(*Substitute* group) allow volumetric navigation through the superimposition of a 3D virtual image onto the patient's head, while systems with no tracking and manual registration (*Affordable* group) perform only a superimposition of a 2D image onto the patient's head. For this reason, tracking systems are more complete and have an easier registration method than non tracking systems. On the other hand, non tracking solutions require less equipment, making them more affordable and easier to adopt. Low cost and easy adoption are important not only for future research but also for employing the technique in hospitals with less resources.

The solution implemented in this work, ARNeuro, does not provide CT/MRI superimposition like the analyzed systems. Instead, we have limited the functionality to only show the center of the incision region, overlaid onto the patient's head, on the display device. The rationale behind this is that incision and craniotomy planning do not require advanced AR visualizations, but a simple system that accurately indicates where the incision has to be performed. This way, we combined the best features of the systems presented in this review. We provide a mobile application with the following characteristics:

- **Affordable**: Executes on high-end smartphones or tablets, that do not cost more than US$ 1,000.00
- **Tracking**: Performs tracking using the device camera and inertial sensors, without the need of additional hardware, like other cameras or a PC
- **Image markers registration**: Fast and simple registration using 3 image markers placed on the patient's head
- **Good accuracy**: Latest version has an overall mean TRE < 3.1mm, from our tests

Comparing with the non-tracking systems in Table 5, our solution does not need so many steps, i.e. preprocessing the virtual image, taking specific pictures of the patient, or manual registration. Also, the device does not need to be in a fixed position.

Comparing with the optical tracking systems in Table 6, our application is standalone, so it has no lag in the video feed, and it does not require any additional hardware (e.g. cameras or PC) in the OR. This approach also does not require cables, connection to other devices, or trackers attached to the mobile device. All these characteristics make the system more affordable, and easier to use and to test in different environments.

Even though the current implementation is not suitable for intraoperative navigation, it can be a good solution for the initial macroscopic part of the surgery, when microscope-based AR systems are not ergonomically viable. This means that this system can be an additional tool for the neurosurgeon when a neuronavigator is available, or a low-cost alternative to a neuronavigator in a hospital with few resources.

3

Theory

This chapter explains the basic concepts of the Computer Vision (CV) techniques applied in this work. Most of the techniques are implemented in a framework called ARKit (APPLE INC., 2018a), developed by Apple, and the internal details are not publicly available. Therefore, the intention of this chapter is not to provide a deep understanding of the presented topics, but instead to give a useful background that helps to understand the system implementation and its limitations.

3.1 Software frameworks and ARKit

A software framework is an abstraction that provides particular functionality to facilitate the development of software applications. Similar to a software library, a framework provides abstraction and code reuse, which allows the developer to rely on previous work that was tested and optimized, without the need of dealing with the internal details or "reinventing the wheel". For example, AR frameworks usually provide an easy way to retrieve the device location in the environment, abstracting all the complex algorithms required for this task.

In this work, we decided to use an AR framework for iOS devices, i.e. iPads and iPhones, called ARKit (APPLE INC., 2018a), developed by Apple. ARKit provides high-level functions that abstract the implementation of many CV techniques. The advantage of using an AR framework is that it is possible to focus on solving the problem, i.e. finding the incision region, instead of implementing CV techniques that have already been implemented. In addition, ARKit is developed and maintained by Apple, which guarantees that it is optimized to make the best use of Apple's hardware. As Miesnieks (2017) points out, this hardware tuning is really important for the good performance of an AR system, and since ARNeuro requires millimetric precision and was developed for real clinical use, ARKit seemed to be the right choice.

The main disadvantage of using ARKit is that its source code is not publicly available. This means that sometimes we have to "guess" what is being done internally, and also that it is not possible to make fine-tuning adjustments to the code. The way to mitigate the "guessing" problem is to read ARKit documentation (APPLE INC., 2018a), watch the video about the inner work of the framework (APPLE INC., 2018b), perform many tests, and understand the employed CV techniques, which are briefly described in the next sections. Since making code adjustments is impossible, an alternative is to create functions that execute in parallel with ARKit, but this can be tricky, because ARKit uses many hardware resources. For example, ARKit 1.5 only uses one camera, but it does not allow the activation of the other camera while it is executing. The use of the Graphics

Processing Unit (GPU) is also limited, so some heavy image processing algorithms may not work properly.

The other big disadvantage of using an AR framework is that the framework updates may break existing code. For instance, the first version of ARNeuro had a color threshold filter that executed in parallel with ARKit 1.0, but this filter stopped working after we updated ARKit to version 1.5.

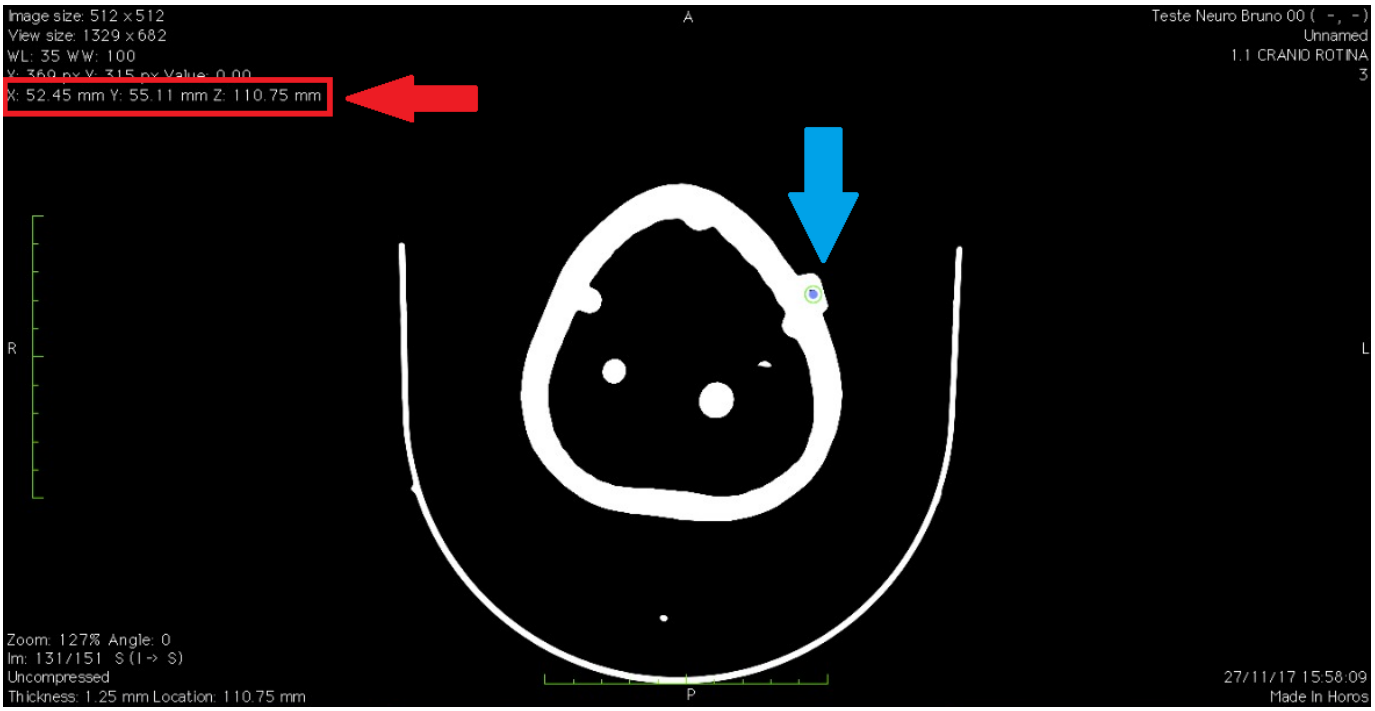
It is worth mentioning that the approach presented in this work could be implemented for other systems, e.g. Android, using a different framework, like ARCore (GOOGLE LLC, 2018). There are also other AR frameworks that could be used, like Vuforia (PTC INC., 2018) or Wikitude (WIKITUDE GMBH, 2018), to name a few. We chose to use ARKit because of the available hardware, i.e. iPhone 7 Plus, and because it is free.

3.2 Coordinate systems

There are three important coordinate systems for our application: the image coordinate system (ICS), the screen coordinate system (SCS), and the world coordinate system (WCS). Besides the coordinate systems, there is also the physical space, which is the real world.

The ICS is a 3D coordinate system of the medical imaging viewer software. This software is not part of ARNeuro, and is usually already available in hospitals. The first step of our approach requires that the neurosurgeon marks the registration points and the target point on the virtual image, using an imaging viewer software. When each point is marked, the software informs its position, which is indicated by a 3D point $p = (x, y, z)$, where $x, y, z \in \mathbb{R}$, in the ICS. Figure 15 shows a screenshot of the free medical image viewer software Horos (HOROS PROJECT, 2018). The blue arrow indicates the point marked by the neurosurgeon, while the red arrow and the red box on the top-left indicate the 3D coordinates of the marked point in the ICS.

Figure 15 – Screenshot of the medical imaging viewer software Horos (HOROS PROJECT, 2018). The blue arrow indicates the point marked by the neurosurgeon. The red arrow and the red box on the top-left indicate the coordinates of the marked point in the ICS.

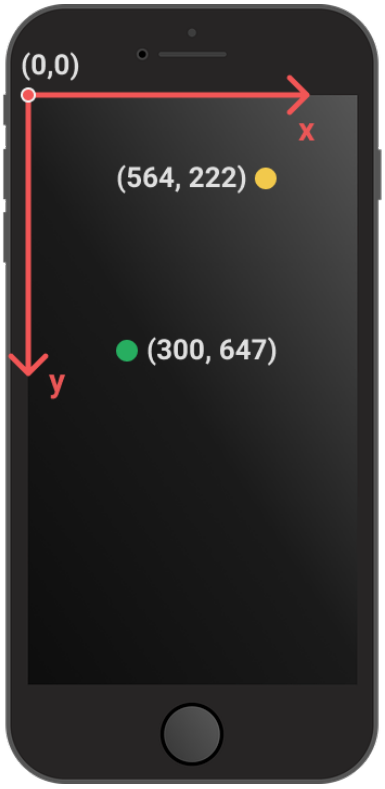


The SCS is a 2D coordinate system of the device screen. Considering the device on portrait orientation mode, the origin of the SCS is at the top-left of the screen. The positive x-axis points to the right of the screen,

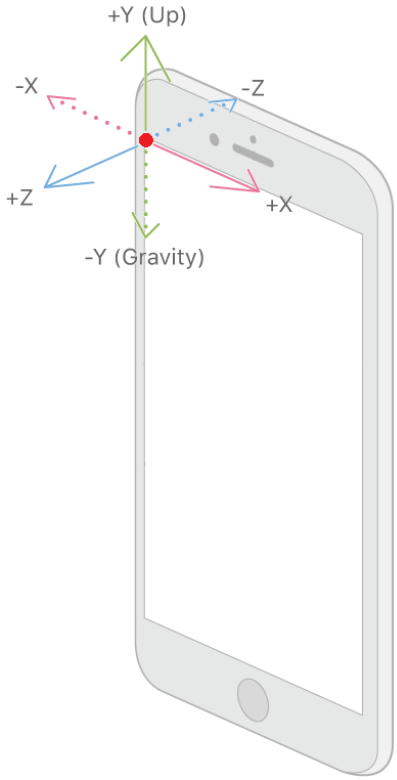
while the positive y-axis points to the bottom of the screen. A position in the SCS is indicated by a 2D point $p = (x, y)$, where $x, y \in \mathbb{N}$. Figure 16a shows the origin and the axes orientation of the SCS.

The WCS is a 3D coordinate system that serves as a common ground between all the objects in the AR environment. This means that objects represented in ICS or SCS have to be transferred to WCS before they can be displayed in AR. These transformations between coordinate systems are explained in chapter 5. In this work, the WCS is determined when the registration step begins. The origin is the initial position of the device. The negative y-axis points downward, based on the gravity, detected by the device's motion sensing hardware. The negative z-axis points in the direction of the device back camera, and is perpendicular to the y-axis. The x-axis is orthogonal to the other two axes and follows the right hand rule, so it points to the right (for a viewer looking in the negative-z direction). A position in the WCS is indicated by a 3D point $p = (x, y, z)$, where $x, y, z \in \mathbb{R}$. Figure 16b shows the axes orientation of the WCS.

Figure 16 – Coordinate systems. **(a)**: World coordinate system. The positive y-axis points upward. The other axes are relative to the device initial orientation. **(b)**: Screen coordinate system. The origin is at the top-left. The positive x-axis points to the right of the screen, and the positive y-axis points to the bottom of the screen.



(a)

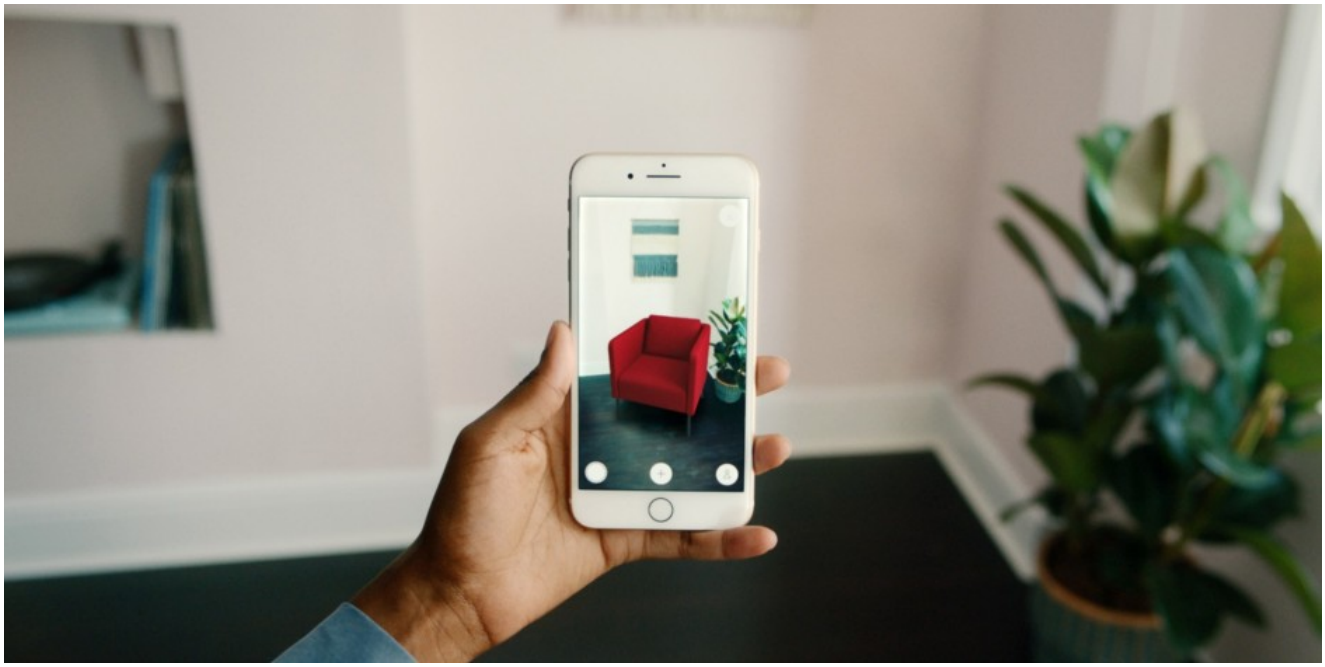


(b)

Source Apple Inc. (2018a), modified

The physical space, i.e. the real world, is a three dimensional environment that is unknown to the application before being observed by the camera. When this environment is observed, ARKit creates an internal representation of it in the WCS, using a technique explained in section 3.3. Since the observed physical space is represented in WCS, it is possible to interact with the real world using WCS coordinates. For example, it is possible to place objects that seem to be in the real world, but are actually in the AR environment, as shown in Figure 17.

Figure 17 – Virtual chair in AR environment.



Source – Eskenazi (2018)

3.3 Visual inertial odometry

Visual Odometry (VO) is a technique that estimates the motion of a visual sensor, i.e. a camera, using only its visual input (NISTÉR; NARODITSKY; BERGEN, 2004).

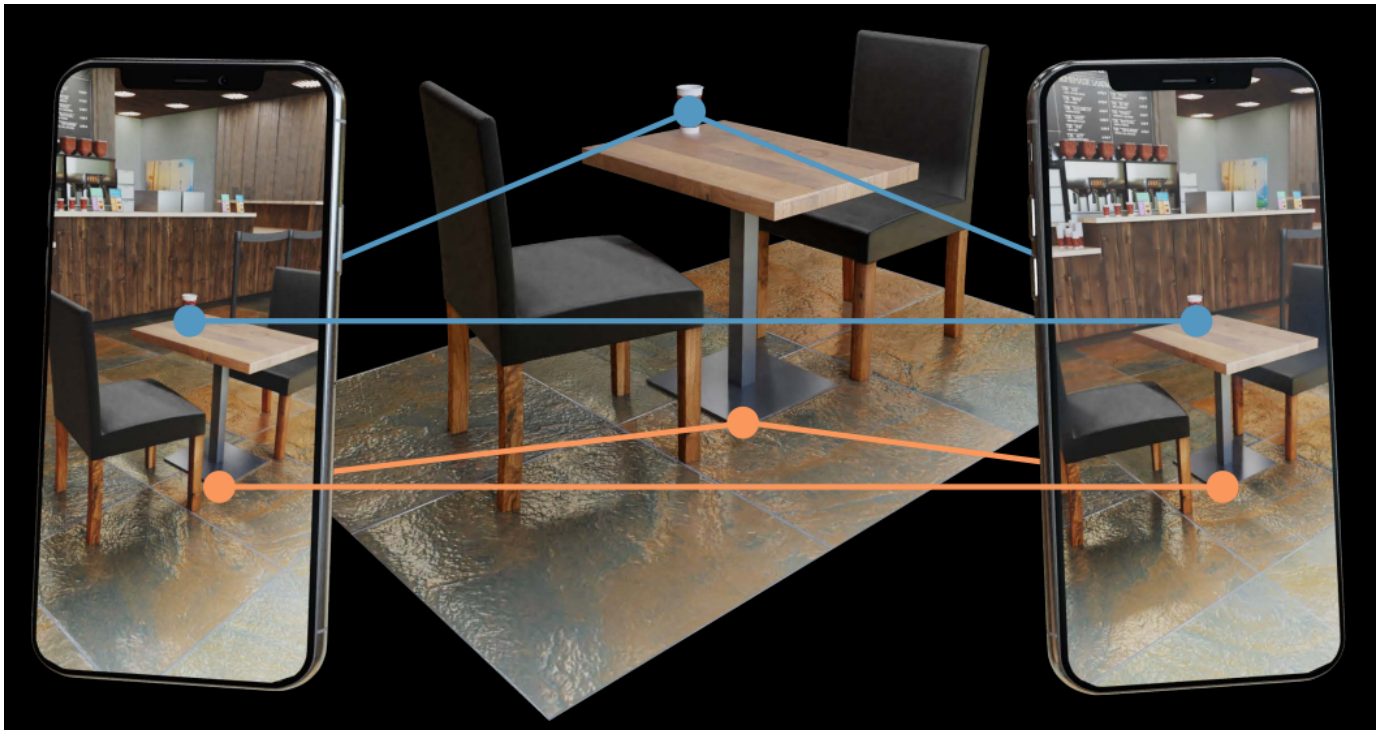
The initial step of VO is detecting "interesting regions" in the image. These regions, also called features, are unusual parts of the image, which can be easily identified across different images of the same scene but obtained from different points of view. Figure 18 depicts this situation, where the device on the left shows the same scene as the device on the right, but from a different position. The features are the blue and orange dots.

In a high-level description, the idea is to match the detected features between pairs of frames and perform a 3D triangulation of the matched points (Figure 18). This triangulation is used to extract depth information and estimate the camera pose. The mathematical background behind this is called Epipolar Geometry and a good explanation is presented by Hartley & Zisserman (2003). VO also relies on other algorithms, like Random Sample Consensus (RANSAC) (FISCHLER; BOLLES, 1987). A complete explanation of VO is given by Nistér, Naroditsky & Bergen (2004).

Visual Inertial Odometry (VIO) combines the visual information obtained through VO with inertial information obtained from an inertial measurement unit (IMU). This combination became popular in mobile robotics because it offers complementary characteristics and helps to improve the system accuracy (LEUTENEGGER et al., 2015).

In our case, the IMU is composed by an accelerometer and a gyroscope, both available in the device. The motion data provided by the IMU is updated on a high frequency, about 1000Hz (MIESNIEKS, 2017). Since this data is not totally precise, the errors quickly accumulate (APPLE INC., 2018b). For this reason, IMU measurements are good for small time intervals, but cannot be used alone to determine the precise location of the device. In order to compensate the error of the inertial measurements, VO is applied, and its information is combined with the motion data, making it VIO. This gives higher accuracy to the system, at the cost of

Figure 18 – Visual Odometry. The features (blue and orange dots) are detected in the images and matched. The 3D triangulation allows the estimation of the camera pose.



Source – Apple Inc. (2018b)

computation time (APPLE INC., 2018b). Currently, the camera frames are obtained only from one camera, so it is important to translate the device to provide enough parallax for the 3D triangulation shown in Figure 18. The visual measurements are obtained from the camera frames, so the update frequency should be no higher than 60Hz, as this is the camera frame rate. Since the update frequencies of the motion data and the visual data are very different, precise clock synchronization between the IMU and the camera is fundamental to obtain good results in VIO. Some manufacturers have started to include a synchronized sensor hub for the hardware components in modern mobile devices, making VIO more viable, with appropriate sensor calibration (MIESNIEKS, 2017). Without this hardware synchronization from factory, it is the job of the developer to implement some steps to perform this synchronization.

To summarize, VIO takes higher accuracy for large time intervals from the visual measurements, and takes higher update rates and good accuracy for small time intervals from the inertial measurements. Translating the device is important to perform the VO part.

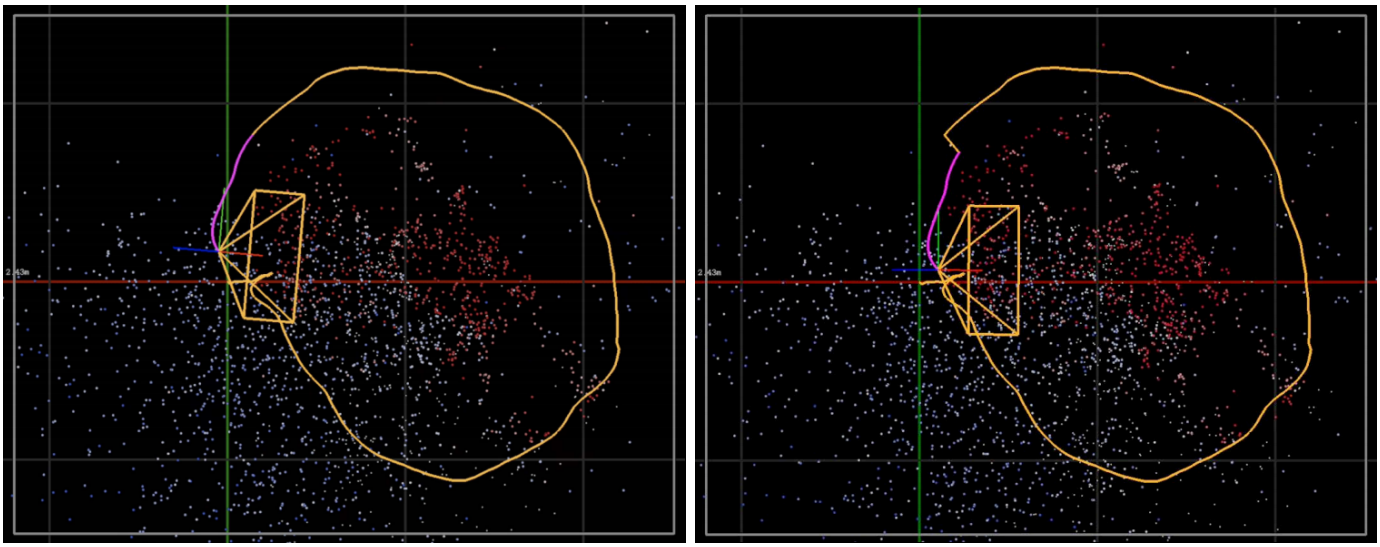
3.4 Simultaneous localization and mapping

Simultaneous localization and mapping (SLAM) is the problem of incrementally building a map of an unknown environment while simultaneously keeping track of the agent's location (DURRANT-WHYTE; BAILEY, 2006).

In this work, when the registration step begins, the application sets the WCS and starts the VIO, as explained in section 3.2 and section 3.3, respectively. While the device is being moved around by the user, its tracking position is updated and the VIO detected features are used to create an internal virtual map of the physical environment. During this time, though, some errors may be accumulated. For this reason, if the camera

returns to a similar view it has seen before, the AR framework performs an optimization, aligning its internal map with the real physical environment that is currently being observed, and correcting the device position. This optimization is shown in Figure 19. The device position update can be clearly seen, while the internal map update is harder to spot.

Figure 19 – Optimization step performed by ARKit. When the camera returns to a similar view it has seen before, the framework updates its internal map of the environment and the device position from **(a)** to **(b)**. The yellow contour is the long-term position tracking, while the magenta is the short-term position tracking, which is updated.



(a) (b)

Source – Apple Inc. (2018a)

4

System requirements

Currently, AR systems that assist in neurosurgery procedures are either hard to use and require many preoperative steps, or hard to adopt and expensive. An approach that is accurate, affordable, easy to use, and easy to adopt is needed. In this chapter, we present this problem and break it into smaller subproblems. In the following sections, these subproblems are introduced as high-level descriptions of the non-functional and the functional requirements for this new system. This way, it is possible to understand what to expect from the application, and what was required to implement it and to solve the problem.

4.1 Non-functional requirements

In the Introduction (chapter 1), it is mentioned that neuronavigation systems are expensive and require a high cognitive load from the neurosurgeon. Many AR approaches have been suggested as neuronavigators replacements, but most of the alternatives are hard to adopt or hard to use. In addition, craniotomy planning is considered one of the most important parts of the neurosurgery, but the current tools could have better usability. For those reasons, the proposed system non-functional requirements are:

- **Accurate**: Mean TRE $<$ 5mm, which is considered good in literature (HOU et al., 2016).
- **Affordable**: The overall approach must cost less than a neuronavigator or an AR neuronavigation system.
- **Good usability**: The software must: (a) require less time to perform registration than the solutions evaluated in our literature review; (b) not need many preoperative steps.
- **Easy to adopt**: The application should be distributable through the internet, and the overall approach must not require additional software or hardware, besides the image markers, and the medical imaging viewer software, such as Horos (HOROS PROJECT, 2018), which is usually already available in the hospital.

4.2 Functional requirements

From our systematic review (chapter 2), we found that the main limitation of the systems that did not perform tracking was that they required many preoperative steps. On the other hand, the systems that performed

optical tracking relied on specific cameras and also needed a desktop to process the information. With the goal of avoiding the burden of many preoperative steps and the dependency on extra hardware or software, the standalone mobile application has the following functional requirements:

- **Request registration points coordinates**: The application must request from the user the coordinates of the target point and the virtual registration points, named Point A, Point B, and Point C.
- **Detect and classify physical registration points**: The application must detect the 3 physical registration points, which are indicated by image markers placed on the patient's head. Each marker must be classified by the application as Point A, Point B, or Point C, after it is detected, to represent its respective virtual registration point.
- **Determine the coordinates of each physical registration point in the WCS**: The application must determine the position of each detected physical registration point in the WCS.
- **Perform the registration**: The application must perform the best alignment of the virtual registration points with their respective physical registration points.
- **Draw incision point**: The application must draw, on the device screen, the incision point overlaid on the patient's head.

4.3 AR challenges

As mentioned in the Introduction, AR has some inherent challenges, namely occlusion, depth perception, and inattentional blindness. ARNeuro does not have to deal with these challenges yet because the current purpose of the application is to indicate the location of the incision region, not to provide a complete neuronavigation system. Even though these issues are not explored in this work, they are briefly discussed here because future versions of ARNeuro may have these AR challenges.

Occlusion is the capability of hiding virtual objects behind real objects. If occlusion is not properly handled, it can be a problem in the intraoperative step, because the virtual image (CT/MRI) superimposed on the patient's head can occlude real objects, e.g. the medical instruments, and the surgeon's hands. As explained by Guha et al. (2017), there are some studies that propose ways to create occlusion using techniques that perform edge detection and color-specific surfaces in the camera feed. ARNeuro does not have an occlusion problem, because just one virtual point is rendered to indicate the position of the center of the incision region, and this is a preoperative step.

Depth perception is how well the user can perceive the distances of the virtual objects. This also is a problem that usually appears in the intraoperative step, because the surgeon needs to correctly perceive the depth of the tumor and other regions of interest. One way to handle this problem is to use different colors to indicate different depths (GUHA et al., 2017). As shown in section 6.2, in ARNeuro the depth can be easily checked if the mobile device is positioned on a sagittal position relative to the patient.

Inattentional blindness is the inability to notice an object because the attention is focused on another object or task. Like the other AR challenges, this can be an issue during the surgery, but not in the preoperative step, for which ARNeuro was designed. Guha et al. (2017) state that some works suggested wire-mesh and "inverse-realism" overlay techniques, rather than solid overlay, to potentially reduce inattentional blindness.

5

Material and methods

This chapter describes the usage and internal working of ARNeuro. First, we present the material used in the development and testing of the software. Second, we introduce ARNeuro concepts to help understanding the different sets of points that are mentioned in this chapter. Third, it is described how our approach finds the center of the incision region in a patient. In each step, we explain the user-level interaction, and how the software solves the problem internally. Finally, it is detailed how we performed the tests to validate ARNeuro.

5.1 Material

Throughout this chapter, two versions of ARNeuro are mentioned: **Old ARNeuro**, and **Current ARNeuro**. Old ARNeuro was developed with ARKit 1.0. Current ARNeuro was developed with ARKit 1.5. If just **ARNeuro** is mentioned, it means that it applies to both versions.

Besides both versions of ARNeuro, the material used in our experiments were: a common PC with the medical image viewer software Horos (HOROS PROJECT, 2018) installed, three different models of the iPhone (Apple Inc., Cupertino, CA, USA), ARNeuro image markers, a styrofoam sphere with 20cm of diameter, a phantom head based on a real patient, and the CT of the phantom head (Figure 20).

Figure 20 – Material. **(a)**: Styrofoam sphere with ARNeuro PRP placed on it. **(b)**: Phantom head. **(c)**: Virtual image of the phantom head.

(a) (b) (c)

5.2 ARNeuro concepts and abbreviations

In section 3.2, the coordinate systems and the physical space are explained. In this section, to facilitate the understanding of the mappings between these coordinate systems, we define some groups of points, and explain their relationship. The abbreviations introduced in this section are extensively cited in the next sections, and they are used interchangeably to indicate just one point or the entire group.

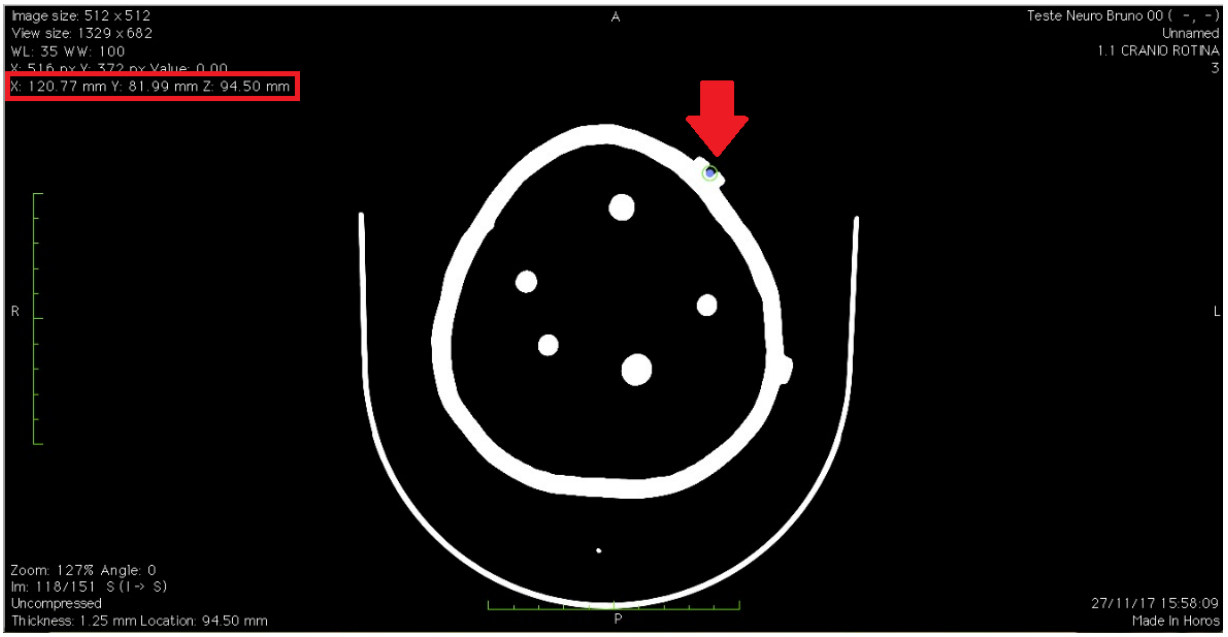
- **Virtual registration points (VRP)**: *VIRTUAL* points identified by the neurosurgeon on the virtual image of the patient using a medical imaging viewer software. Initially, these points are in the ICS. Figure 21a shows how the neurosurgeon marks these points. Later, these points are transferred to the WCS, so ARNeuro can perform the registration. Figure 21c shows one VRP (blue point), drawn by ARNeuro after the registration.
- **Physical registration points (PRP)**: *PHYSICAL* markers placed on the patient's head to indicate the correspondent locations of the VRP. These are the **center of the image markers**, shown in Figure 21b. These points are in the physical space.
- **AR registration points (ARRP)**: *VIRTUAL* points that represent the PRP in WCS. When a PRP is recognized by ARNeuro, an ARRP is created to represent that PRP in WCS. Figure 21c shows one ARRP (red point), drawn by ARNeuro.
- **Physical target point (PTP)**: *PHYSICAL* point that indicates the real location of the incision, marked on the phantom head, in the physical space. This point is only used to validate our approach, and it is not going to exist in a real clinical scenario. In our tests, the PTP was marked at the center of a piece of millimeter paper (Figure 21d).
- **Virtual target point (VTP)**: *VIRTUAL* point that indicates the location of the incision, inferred by ARNeuro. As the VRP, the VTP is initially in the ICS, and is later transferred to the WCS. First, it is marked by the surgeon on the image viewer software (Figure 21a). After the registration is performed, this point is drawn by ARNeuro, to indicate the incision place. Figure 21d shows the AR representation of the VTP (yellow point).

5.3 Steps to find the incision point - Overview

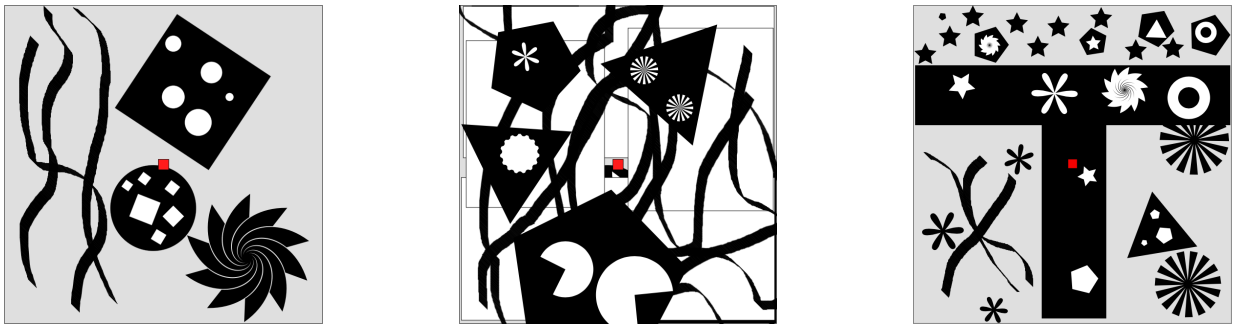
Figure 22 shows each step mentioned in this section. The groups in purple do not appear on screen. The other groups appear in their respective color. The details are described in the next section.

1. The user informs the VRP and the VTP to ARNeuro. PRP are marked on the patient's head.
2. PRP are recognized by ARNeuro and represented as ARRP in the WCS.
3. VRP is moved to WCS and the registration is performed to best match VRP with ARRP.
4. VTP is moved to WCS. The mapping found in the registration is used to transform VTP to the center of the incision region.
5. The distance between the VTP and the PTP is calculated to analyze the system accuracy.

Figure 21 – Groups of points used by ARNeuro. **(a)**: Medical imaging software screenshot. The red arrow indicates one VRP, marked by the surgeon. The red box on the top-left indicates the VRP coordinates, in ICS. This is also how the VTP is obtained. **(b)**: 3 image markers, used to indicate the PRP. **(c)**: A PRP detected by ARNeuro. The red point is an ARRP, correctly placed by ARNeuro, at the center of the PRP. The blue point is a VRP, which **should** be aligned with the ARRP after the registration. **(d)**: Millimeter paper with the PTP at the center. The yellow point is the VTP, inferred by ARNeuro.

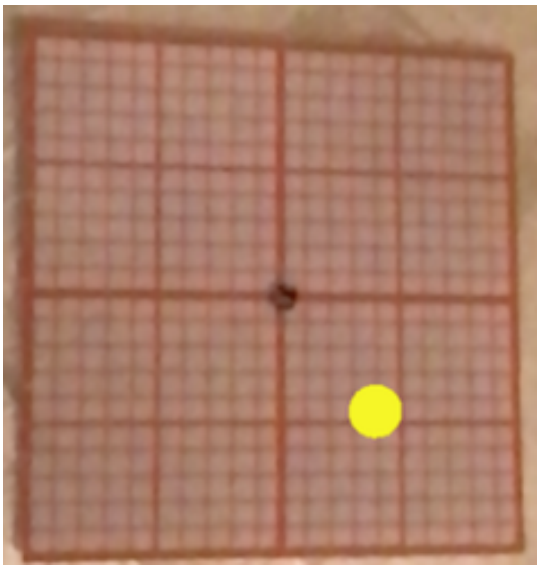


(a)



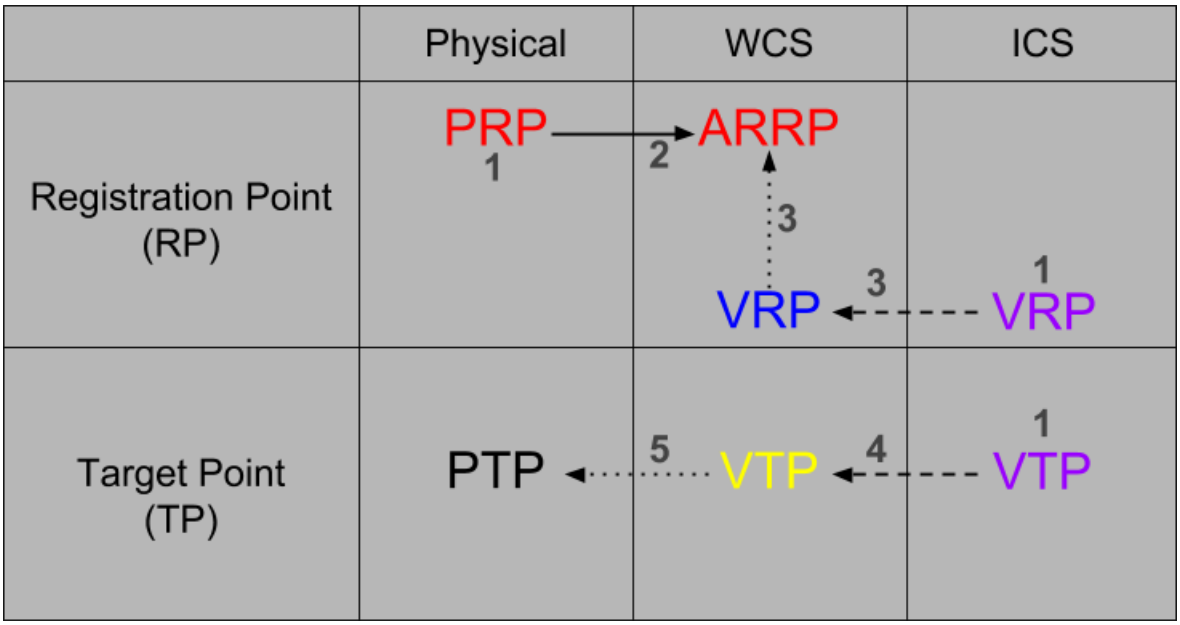
(b)

(c)



(d)

Figure 22 – Groups of points represented in different coordinate systems. The solid line indicates the representation of a group into another coordinate system. The dashed line indicates a geometric transformation that changes the group coordinates to another. The dotted line represents a comparison of the locations of those groups. Steps: **1-** VRP and VTP are informed to ARNeuro by the user. PRP is marked on the patient. **2-** PRP are recognized by ARNeuro and represented in ARRP. **3-** After creating the 3 ARRP, VRP changes from ICS to WCS and the registration is performed, to best match VRP with ARRP. **4-** VTP is transferred to WCS, and the mapping found in the registration is used to transform VTP to its correct position, which is the center of the incision region, inferred by ARNeuro. **5-** The distance between the VTP and the PTP is measured to find the error.



5.4 Finding the incision point

This section describes how to use ARNeuro to find the incision point, in a user-level, and also what the application is doing internally to solve the problem. The implementation differences between Old ARNeuro and Current ARNeuro are mentioned as a way to improve the discussion on how to solve the problem. In addition, some of the previous approaches may be combined with the current ones in future work.

Each of the following sections is related to a specific functional requirement.

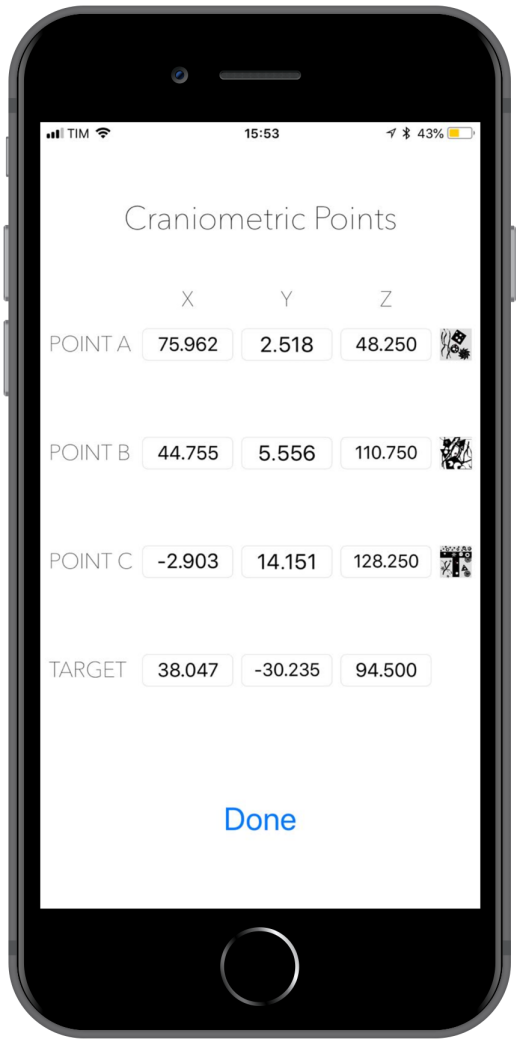
5.4.1 Inform required points coordinates

The first step requires the user to type the coordinates of the VRP and the VTP, in millimeters. These coordinates are obtained from the patient virtual image, opened with a medical imaging software, as explained in section 5.2. Figure 23 shows the ARNeuro screen where the user informs the coordinates.

For the VRP, the neurosurgeon can choose any 3 non-collinear points that are easily identifiable in the virtual image and on the patient's head. The reason for needing easy identification is that the VRP have to be marked on the patient's head with image markers, which indicate the PRP. In our tests, for example, any fiducial on the phantom could be chosen as a VRP, because they are easily identifiable in the physical space (Figure 20b) and in the virtual image (Figure 20c, Figure 21a).

The surgeon also has to inform the VTP. This point can be any point that the surgeon wants to find on the patient's head, but usually it's going to be the center of the incision region.

Figure 23 – First screen of ARNeuro. The neurosurgeon must type the coordinates of the VRP (POINT A, POINT B, POINT C), and the VTP (TARGET).



5.4.2 Detect and classify the PRP

In order to detect the PRP, the only thing that the user has to do is to approximate the mobile device camera close enough to the PRP, so they can be recognized by ARNeuro. To the user, this is the start of the registration step, which finishes when the last PRP is detected and the incision point is shown on screen. Old ARNeuro detected the markers by their colors, while Current ARNeuro uses image recognition.

5.4.2.1 Old ARNeuro

In the initial versions of ARNeuro, which used ARKit 1.0, a simple image segmentation technique was implemented to identify the PRP by their colors. In our experiments, we found that blue and green markers gave good results, while red markers were not a suitable choice due to the presence of red in white skin tones. The image segmentation can be summarized in 6 steps:

- Step 1: Convert camera frames to CIImage objects

  The implementation of the ARSessionDelegate protocol gives access to the frames captured by the device camera through a function called *session(_:didUpdate:)*. A camera frame is an ARFrame object that contains a CVPixelBuffer object, which is the view of the world projected by the camera. CVPixelBuffer object should be converted to a CIImage object, because this is required to use a CIFilter.

- Step 2: Pass the CIImage object to a CIFilter object

  The CIImage object has to be passed to a CIFilter object. Even though there are many built-in filters already implemented, we did not find any that performed exactly the operations that we wanted. For this reason, we developed a custom filter class called ThresholdFilter, which is a subclass of the CIFilter. Subclasses of CIFilter need a CIKernel object. A CIKernel is a wrapper that contains the filter code, written in Core Image Kernel Language, which is a dialect of the OpenGL Shading Language (GLSL).

  The main reason for using a CIFilter is to optimize the image processing, making use of the GPU instead of the CPU. Steps 3 and 4 were implemented as GLSL functions of the same CIKernel.

- Step 3: Convert the 2D image of the camera frame from RGBA color space to HSV color space

  This step is done because it is more intuitive to perform color threshold in HSV than in RGBA. Hocevar (2013) developed a GLSL implementation of the RGBA to HSV conversion algorithm to make the best use of the GPU (Source code 1). We used this code in ThresholdFilter's CIKernel.

Source code 1 – RGBA to HSV color space conversion in GLSL.

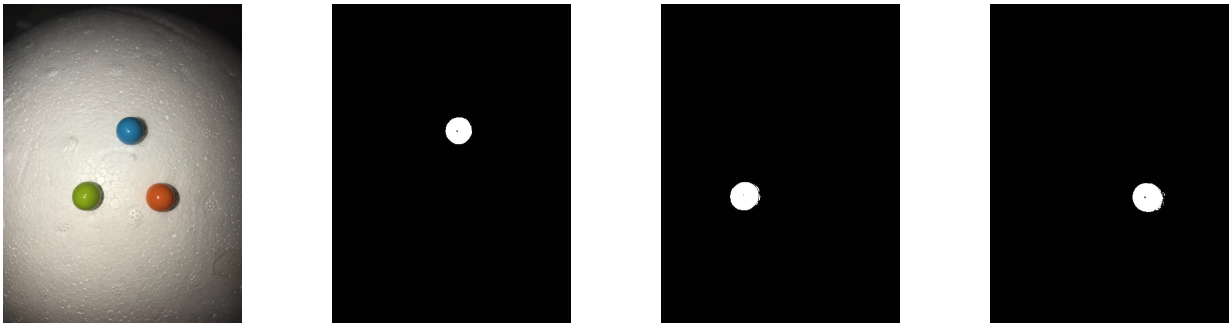
```
vec3 rgb2hsv(vec3 c) {
    vec4 K = vec4(0.0, -1.0 / 3.0, 2.0 / 3.0, -1.0);
    vec4 p = mix(vec4(c.bg, K.wz), vec4(c.gb, K.xy), step(c.b, c.g));
    vec4 q = mix(vec4(p.xyw, c.r), vec4(c.r, p.yzx), step(p.x, c.r));

    float d = q.x - min(q.w, q.y);
    float e = 1.0e-10;
    return vec3(abs(q.z + (q.w - q.y) / (6.0 * d + e)), d / (q.x + e), q.x);
}
```

Source – Hocevar (2013)

- Step 4: Perform a color threshold operation on the image

  After converting the pixel from RGBA to HSV, ThresholdFilter applies the color threshold operation, which basically consists in checking if the pixel Hue is in the interval that corresponds to the color we want to identify. If the pixel Hue is in the interval, that pixel is converted to the white color; else, it is converted to the black color. See Figure 24.

Figure 24 – Color threshold, the first approach of identifying the PRP. **(a)**: The colored PRP. **(b)**: Blue threshold. **(c)**: Green threshold. **(d)**: Red threshold.



(a) (b) (c) (d)

In order to identify the green color, for example, we obtained good results with $0.45 < \text{Hue} < 0.61$, and Saturation > 0.6. See Source code 2 for an example of the green threshold.

Source code 2 – Green color threshold. All the values are float numbers between 0.0 and 1.0.

```
vec4 green_threshold(vec3 c) {
    vec3 hsv = rgb2hsv(c);
    if ((hsv.x > 0.45) && (hsv.x < 0.61) && (hsv.y > 0.6))
        return vec4(1.0, 1.0, 1.0, 1.0);
    return vec4(0.0, 0.0, 0.0, 0.0);
}
```

In Source code 2, the x coordinate of the vector represents the Hue, while the y is the Saturation, and z is the Value. The values of the if-condition to indicate the green color were determined empirically. In some situations, these values need to be adjusted. This function returns a vec4, i.e. the pixel RGBA channels, because this is required by CIKernel.

- Step 5: Create a binary image bitmap

  The ThresholdFilter object performs the image processing with the CIImage object, but there is a problem. CIImage objects are not actually images. The documentation says that a CIImage is like a "recipe" to create an image, meaning that the image is generated in a lazy fashion, only when the rendering is actually required. For this reason, before the next step, we need to render the image, which is a costly operation. One way of doing this is to create a CGImage object, which is a bitmap of the image. Then, it is trivial to access the image data, as pixels are stored in a raster format.

- Step 6: Identify the centroid of the image

  With the bitmap of the binary image, it is possible to calculate the centroid $c = (x, y)$ of the image, as follows:

$$x_c = \frac{1}{M} \sum_{i=1}^{n} m_i x_i \tag{5.1}$$

$$y_c = \frac{1}{M} \sum_{i=1}^{n} m_i y_i \tag{5.2}$$

  where M is the number of white pixels, m_i is the value of the pixel (1.0 for white, or 0.0 for black), and x_i and y_i are the coordinates of the i-th pixel.

  As explained in (NGUYEN, 2007), it is possible to calculate the centroid using the GPU, but the proposed implementation uses functions that are not available in iOS. For this reason, we decided to leave the GPU implementation for a later optimization, and, to validate our approach as soon as possible, we chose to implement the centroid code in Swift, to execute on the CPU. The input of the function is a row-major vector composed by 0s and 1s, which represents the binary image, the height, and the width. Source code 3 shows a rough implementation of the centroid code.

Source code 3 – Centroid calculation in Swift language.

```swift
func centroid(rawImage: [Int], height: Int, width: Int) -> (Int, Int) {
    var centroidX = 0
    var centroidY = 0
    var numberOfWhitePixels = 0
    for y in 0..<height {
        for x in 0..<width {
            let index = x + y * width
            centroidY += rawImage[index] * y
            centroidX += rawImage[index] * x
            numberOfWhitePixels += rawImage[index]
        }
    }
    if numberOfWhitePixels > 0 {
        centroidX /= numberOfWhitePixels
        centroidY /= numberOfWhitePixels
        return (centroidX, centroidY)
    }
    return (-1, -1)
}
```

There are two details to keep in mind about this approach.

First, one has to be really careful with the coordinate systems. Core Graphics classes, e.g. CGImage, use a coordinate system which starts at the bottom-left, contrary to the common approach of adopting the origin at the top-left, like other frameworks do, e.g. UIKit. See Figure 25. Source code 3 returns a tuple composed by centroidX and centroidY, which assumes that the coordinate system origin is at the bottom-left of the image, because the data was obtained from a CGImage object. Depending on what classes are used in the project, the coordinates have to be converted to UIKit coordinate system, which is the most used to handle user interactions. UIKit coordinates are similar to the SCS, explained in section 3.2.

Figure 25 – Difference between the coordinate systems of UIKit and Core Graphics.

Source – Apple Inc. (2012)

Second, the steps described in this section are executed when the function *session(_:didUpdate:)* is called. This happens every time the camera frame is updated, which is around 60 times per second, as this is the camera frame rate. In ARKit 1.0, it was possible to execute these steps every time a new frame arrived, but this is not the case with ARKit 1.5. These operations are resource intensive and they start to compete with ARKit for the GPU and the CPU. One common option to handle these situations is to leave the heavy execution on a background thread, but this is not possible, for reasons that are better explained in section 5.4.3.1. In order to use this approach in future versions, there are two optimizations that we can think of: (a) not executing the mentioned steps every time *session(_:didUpdate:)* is called; (b) reducing the image size, which leads to faster computations.

5.4.2.2 Current ARNeuro

The release of ARKit 1.5 beta, in January, 2018, introduced an image recognition feature to the framework. The old approach was then discarded, for two reasons: (a) ARKit image recognition is more reliable; (b) the filter that we implemented to run on the GPU stopped working with ARKit 1.5, probably due to some internal changes in the framework that required more processing power.

Current ARNeuro just loads what images it has to recognize into the framework. For now, the images are the ones from Figure 21b. The application also has to inform the physical sizes of the images to the framework. When an image is detected, ARKit automatically notifies the application what image was detected.

There are some recommendations to use the image recognition feature. For best results, the image has to have the following characteristics (APPLE INC., 2018b):

- High texture
- High local contrast
- Well distributed histogram
- No repetitive structures

Also, if the physical image to be recognized is small, ARKit gives a warning. For example, we are using 15×15mm markers, and the framework shows a warning message stating that a larger physical image would be better. The minimum size is not specified in the documentation but our markers work well. Some tests have shown that ARKit stops giving the warning with physical image sizes larger than 25.4×25.4mm, which is equivalent to 1×1 inch. We also tested the use of 10×10mm markers, but the application could not detect the image. With the current setup, the image can be recognized from a distance of ~4cm.

5.4.3 Determine the coordinates of each PRP in the WCS

As soon as a PRP is detected, ARNeuro creates an ARRP, which is the representation of a PRP in WCS, and displays it on the screen. This process is described in this section.

It is worth noting that, even though the ARRP are the representations of the PRP in WCS, they do not have the same exact locations, because the 3D point retrieved in this step comes from the internal virtual representation of the real world that the framework maintains using its VIO Slam. As it was shown in chapter 3, this internal representation may have errors.

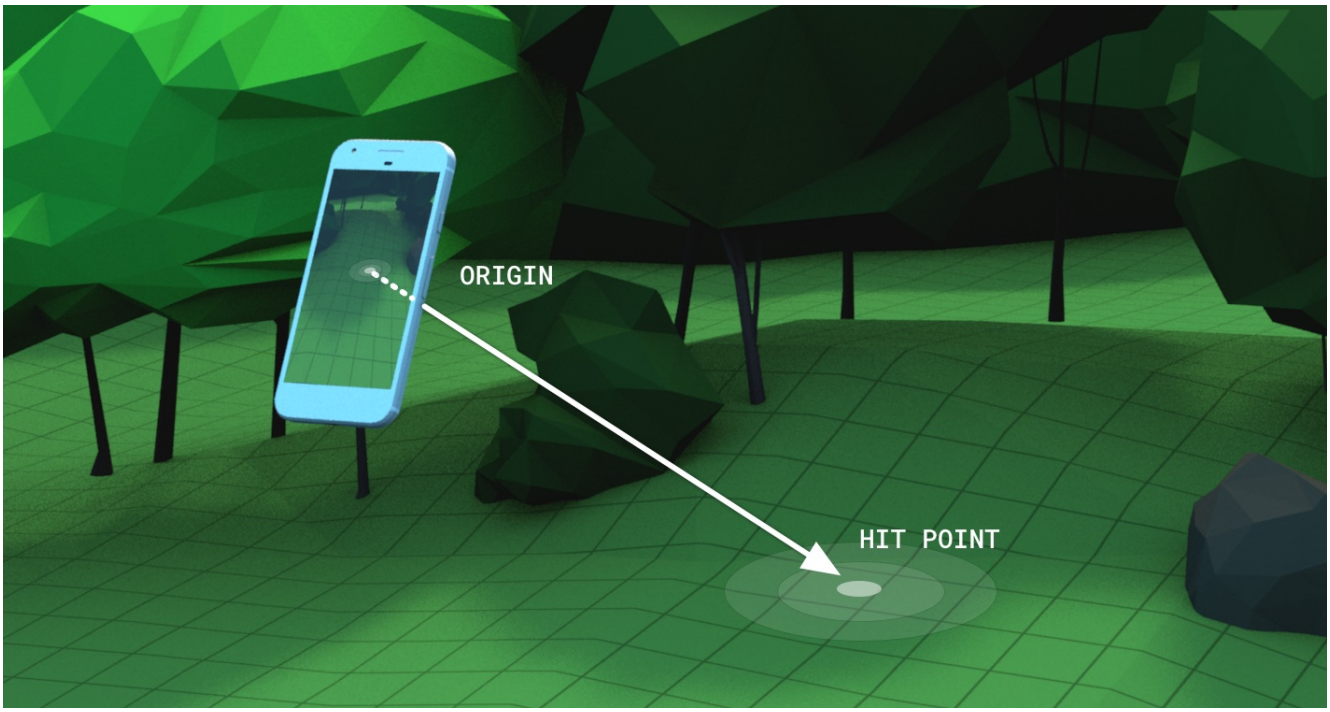
5.4.3.1 Old ARNeuro

As it was mentioned in section 5.4.2.1, when ARKit 1.0 was used, an image segmentation had to be performed to identify the PRP. The centroid of each image was identified, but its representation was in 2D, more specifically in the SCS.

In order to find the incision point, the locations of the PRP in WCS are needed, because we want to provide the best fit of the VRP with the PRP. The problem is that the locations of the PRP in the WCS are not known.

To find a 3D representation of the PRP, there is a class named ARSCNView, which has an instance function called *hitTest(_:types:)* that basically takes as input a 2D point in SCS, and returns a 3D point in WCS. The returned 3D point represents the "hit position" of the first physical object identified by the framework in a line that starts at the device camera and extends in a direction determined by the device orientation and camera projection (Figure 26). The hit position, in the WCS, is retrieved from the knowledge that the framework has about the observed world, and it is contained in an ARAnchor object, in the *transform* property.

Figure 26 – HitTest, a function to convert 2D coordinates in SCS to 3D coordinates in WCS.



Source – Google Inc. (2018)

In Old ARNeuro, the input of the *hitTest* function is the centroid coordinates, in SCS, of the colored marker. For this reason, the steps needed to find the PRP, as explained in section 5.4.2.1, must be done for the same camera frame. In other words, it is not possible to perform those steps in a background thread because, as soon as the steps are done and the centroid is calculated, another frame would have arrived, and the *hitTest* would be called in a different frame as the one that we calculated the centroid.

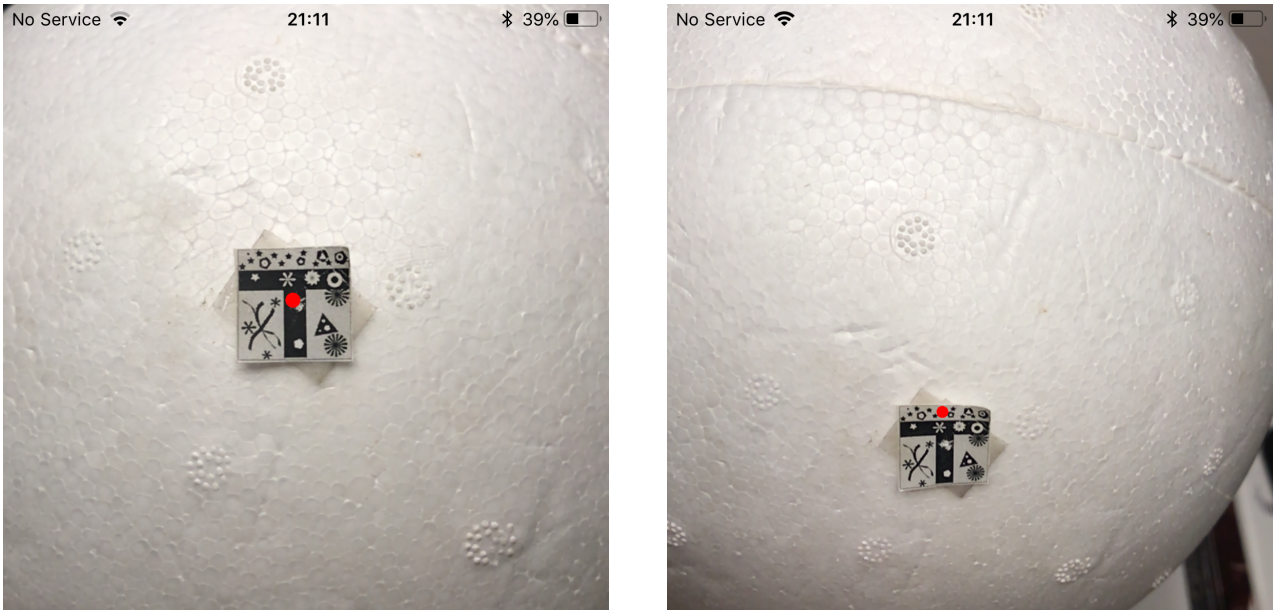
5.4.3.2 Current ARNeuro

With the addition of image recognition, retrieving the ARRP became easier. The protocol ARSessionDelegate should be implemented, so when an image is recognized by ARKit, the function *session(_:didAdd:)* is called, and it informs the ARAnchor of the recognized image. The *transform* property of the ARAnchor object

contains the position of the center of the recognized image. This is better than the other method because, since the framework has previous information about the image, i.e. its size, it can estimate a more accurate position for the ARRP. Also, this approach is less susceptible to noise, because a color filter can suffer from the colors of the environment.

In spite of the better method, this approach also has some issues. The behavior of the virtual points identified by this method is a little bit different than the behavior of points created by the previous method. In the current version, ARKit seems to "force" the AR point to stay in place when the device is facing the image directly, but, when the device is moved, the perceived position of the point in AR changes, seeming as it was not created in the correct place. Figure 27 demonstrates this behavior.

Figure 27 – Behavior of ARRP created by the method described in section 5.4.3.2. **(a)**: The AR point, seen from the initial point of view. **(b)**: The same AR point seen from a different point of view. It moves away from the center of the PRP.



(a) (b)

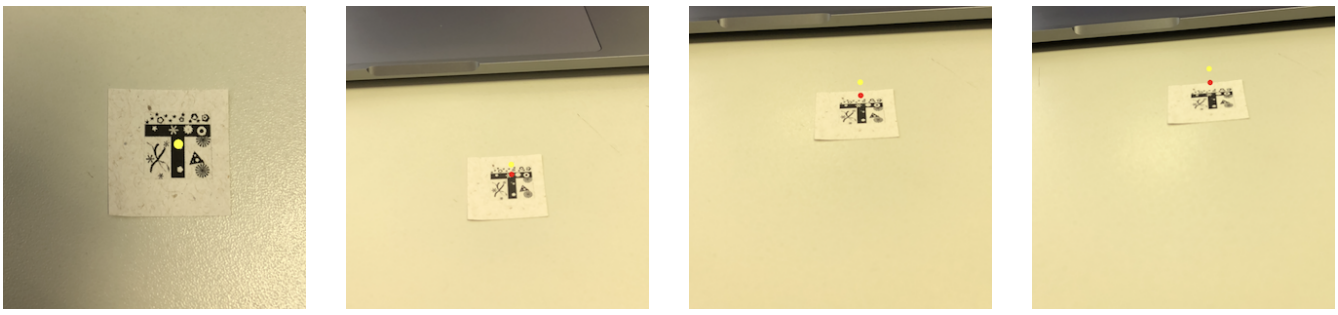
This behavior can have a very negative impact on the application accuracy, because the ARRP are going to appear in incorrect places and affect the surgeon perception of the VTP. Our hypothesis is that this behavior is caused by the image recognition feature uses a special kind of ARAnchor, called ARImageAnchor. For this reason, two short experiments were done to understand what ARKit was doing and how to fix it.

In the first experiment, we tested the default approach of ARKit, using ARImageAnchor. When the image was recognized by the framework, the application added a virtual red sphere at the position of the ARImageAnchor informed by the function *session(_:didAdd:)*. Then, the application performs a *hitTest* function, passing as argument the 2D position, in SCS, of the ARImageAnchor projection. As explained in section 5.4.3.1, the *hitTest* function returns an ARAnchor object, and our goal was to test if the positions of the ARImageAnchor and the ARAnchor were the same, as they should be. A virtual yellow sphere was added at the ARAnchor position of the *hitTest*. The results showed that anchors positions are not the same. See Figure 28a. Also, it seems that, when image recognition is being used, the framework tries to keep the virtual AR objects in correct positions relative to the recognized image. We assume that the problem is that we use small markers, so the framework is not able to recognize the image from certain distances, and then the objects appear to be in the wrong place. For now, it was decided not to use bigger markers because it would be inconvenient

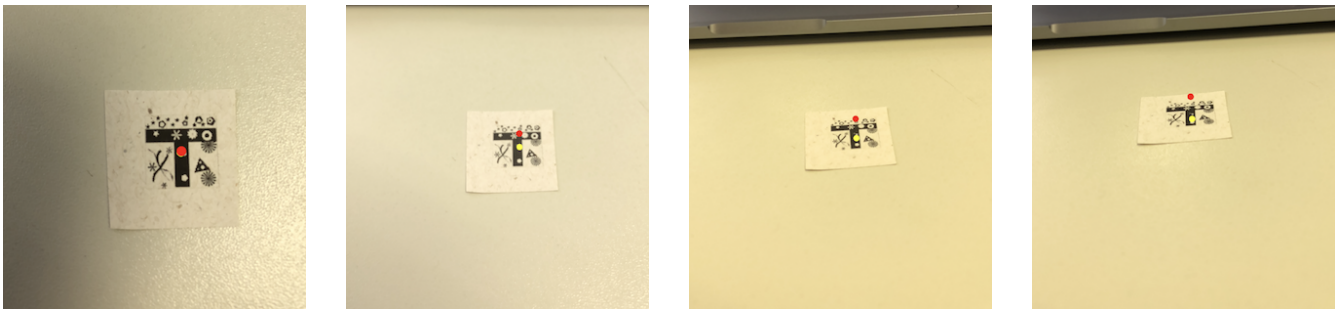
for the surgeon. Another issue is that, since not all markers are observable in the same frame, the framework may try to adjust the AR environment to match the image marker it is observing at the moment. This behavior indicates that the accuracy would be best when the markers are around the incision region, but we do not want this limitation. For this reason, we decided to use another approach, described in the second experiment.

In the second experiment, we eliminated the ARImageAnchor from the equation. When the image was recognized by the framework, we created an ARAnchor with the same position of the ARImageAnchor, and then we removed the ARImageAnchor. In other words, we replaced the ARImageAnchor with a normal ARAnchor. A virtual red sphere was used to represent the position of this ARAnchor. Then, the application performs the same *hitTest* as in the first experiment, and added a virtual yellow sphere at the position of the returned ARAnchor. What was noticed is that the anchors are not at the same position, but the framework stops trying to adjust the virtual objects position based on the image marker. As it is shown in Figure 28b, the yellow sphere stays in the correct place even when the device is in a different point of view. This result suggests that replacing ARImageAnchor objects by ARAnchor objects is a better approach, because it avoids some ARKit "optimizations" that are not desirable in this situation.

Figure 28 – Testing the difference between using ARImageAnchor and ARAnchor. In both images, the yellow sphere is an ARAnchor, returned by the *hitTest* function. In **(a)**, the red sphere is an ARImageAnchor, while in **(b)** the red sphere is an ARAnchor.



(a)



(b)

The conclusions of both experiments are just assumptions, since ARKit is closed source and we had to perform a black-box testing. Also, in order to have more confidence in these findings, more rigorous tests are needed.

5.4.4 Registration

Old ARNeuro did not perform a registration. Instead, the problem of finding the incision was solved using a trilateration, as explained in section 5.4.5.

In Current ARNeuro, the registration starts automatically, when the last of the PRP is detected. There is no specific order for detection. The steps of the registration are as follows:

- VRP and VTP are transferred to WCS. Since the VRP and VTP are in millimeters, and the WCS is in meters, the conversion is simple.
- The registration is performed. The goal of the registration is to find a rotation matrix R and a translation vector T that, when applied to the VRP, gives the best match between the VRP and their respective ARRP.

In order to have a visual feedback of the registration result, ARNeuro draws blue AR points that represents the VRP, so we can see how well they match with the red AR points, which represents the ARRP. An example of bad match is shown in Figure 21c. The application also presents the RMSE of the distances between the VRP and their respective ARRP. Thus, although this error metric seems similar to the Fiducial Registration Error (FRE), we are not measuring the FRE, because the ARRP may not indicate the precise location of the PRP. What is measured is how well the VRP are aligned to the ARRP. The calculated RMSE is presented to the surgeon as an indicator of how well the PRP were placed on the patient's head, assuming that the ARRP have an acceptable accuracy.

Formally, the registration problem is exactly what was described by Arun, Huang & Blostein (1987) in their paper "Least-squares fitting of two 3-D point sets". The problem is defined as follows: Given two 3D point sets $\{p_i\}$ and $\{p_i^{'}\}$; $i = 1,2,...,N$

$$p_i^{'} = Rp_i + T + N_i \tag{5.3}$$

where the p_i and $p_i^{'}$ are considered as 3×1 column matrices, R is a 3×3 rotation matrix, T is a translation vector (3×1 column matrix), and N_i a noise vector, find R and T to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \left\| p_i^{'} - (Rp_i + T) \right\|^2 \tag{5.4}$$

In order to solve this problem, Arun, Huang & Blostein (1987) propose an algorithm that involves the Singular Value Decomposition (SVD) of a 3×3 matrix. This algorithm has 5 steps:

- Step 1: Calculate the centroids c, $c^{'}$ of the point sets $\{p_i\}$, $\{p_i^{'}\}$.
- Step 2: Calculate H, which is a 3×3 matrix

$$H = \sum_{i=1}^{N} (p_i - c)(p_i^{'} - c^{'})^t \tag{5.5}$$

  where the superscript t denotes matrix transposition.

- Step 3: Find the SVD of H,

$$H = U\Sigma V^t. \tag{5.6}$$

- Step 4: Calculate the matrix X,

$$X = VU^t. \tag{5.7}$$

- Step 5: Calculate $det(X)$, the determinant of X, and find the rotation matrix R,

$$R = \begin{cases} X, & \text{if } det(X) = +1 \\ X', & \text{if } det(X) = -1 \end{cases} \tag{5.8}$$

where $X' = V'U^t$, and V' is obtained from V by changing the sign of the 3rd column.

If $det(X) = -1$, there is still a possibility that none of the singular values of H is zero, and then this approach is not appropriate. The complete derivation and explanation can be found in (ARUN; HUANG; BLOSTEIN, 1987). In ARNeuro, the SVD is calculated using the *dgesdd* function, which is part of the Lapack implementation of Apple's Accelerate framework. Finally, it is important to highlight that the solution proposed by Arun, Huang & Blostein (1987) does not work when the points are collinear or when the noise values N_i are too large.

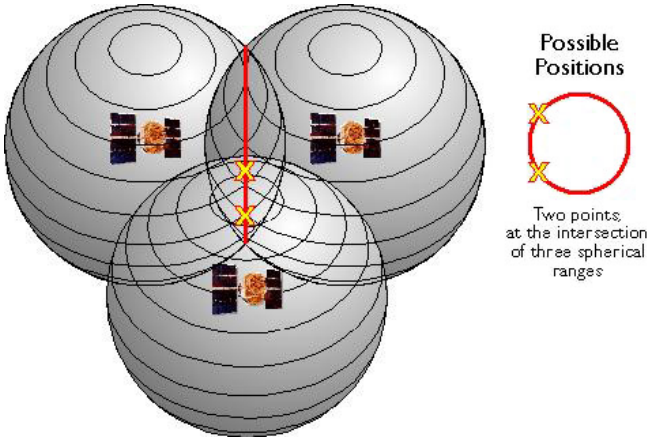
5.4.5 Draw incision point

In this final step, the neurosurgeon can see the incision point on the device screen, overlaid on the patient's head.

5.4.5.1 Old ARNeuro

In previous versions, the problem of finding the incision was not solved by the registration process. Instead, we solved the problem using trilateration, which is a technique commonly used in Global Positioning System (GPS) to find the intersection of three spheres, as seen in Figure 29.

Figure 29 – Trilateration example. Intersection of 3 spheres used in GPS.



Source – DiBiase (2013)

In order to calculate the trilateration, the application performed 4 steps:

- Step 1: Calculate the euclidean distances between each VRP and the VTP

$$d(VRP_i, VTP) = \sqrt{(VRP_{i,x} - VTP_x)^2 + (VRP_{i,y} - VTP_y)^2 + (VRP_{i,z} - VTP_z)^2} \tag{5.9}$$

  where VRP_i is one of the virtual registration points, and VTP is the only virtual target point.

- Step 2: For each VRP, create a sphere with center equal to the respective ARRP position, and radius equal to the VRP distance to the VTP. This created 3 spheres:

$$\text{Sphere } i\text{: } (x - ARRP_{i,x})^2 + (y - ARRP_{i,y})^2 + (z - ARRP_{i,z})^2 = d(VRP_i, VTP)^2 \tag{5.10}$$

  where x, y, and z are coordinates in WCS.

- Step 3: Perform a trilateration to calculate the 2 intersections of the 3 spheres, in WCS.

- Step 4: Eliminate one of the intersections found in step 3.

  To do this, the application calculates the positions of the camera projections of both intersections, in SCS, and then use these 2D positions as inputs for two *hitTest* function calls, targeting the patient's head. The results of the *hitTest* calls are going to be 3D points in WCS that are located on the surface of the patient's head. The intersection which is closer to its respective *hitTest* result is the correct one, because it is closer to the patient's head surface.

This approach is theoretically correct, and the results we present with the styrofoam sphere used this technique. The problem is that the acquisition of the ARRP is not perfect. Therefore, in some situations, if any of the ARRP was not properly acquired, the trilateration process did not find any intersection. This is a problem for using the application in clinical practice, because, if the neurosurgeon misses the correct place of the PRP by some millimeters, the system may not find any incision, and, even if the surgeon perfectly places the PRP, the acquisition of the points may still have errors, and the trilateration may fail.

5.4.5.2 Current ARNeuro

The current solution is straightforward. We already have the VTP, informed by the neurosurgeon in the first screen of the application. The only problem is that the VTP is in the WCS, but it is not correctly positioned yet. However, the "map" for positioning the VTP was found during the registration. Using the rotation matrix R, and the translation vector T, both found in the registration step, the only thing that we need to do is to apply these transformations to the VTP.

Finally, with the VTP coordinates properly adjusted in WCS, the framework draws this point on the screen, overlaid on the patient's head.

5.5 Tests

In this section, it is explained how ARNeuro was tested. Two different experiments were performed. In the first, we used a styrofoam sphere with 20cm of diameter to emulate a human cranium (Figure 20a). Three PRP were placed on the sphere in positions that emulate craniometric points known as *nasion*, above the nose, *prosthion*, under the nose, and *orbitale*, between the cheek bone and the eye. These craniometric points

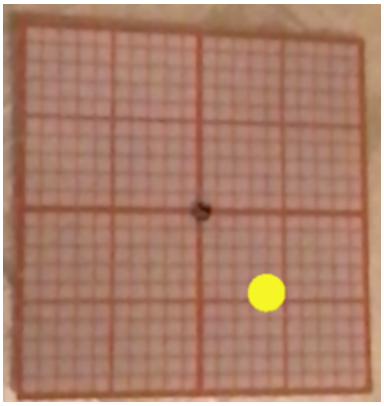
have been chosen because they are easily identifiable landmarks on the head. In the second experiment, the application was tested to find the incision point in a phantom head of a real patient (Figure 20b). The VRP were obtained from the phantom head CT (Figure 20c). We measured the TRE, the registration time, and the registration RMSE.

5.5.1 Target registration error

The information that ARNeuro provides to the surgeon is the center of an incision region, drawn as a virtual AR point. To validate the accuracy of our approach, we calculated the TRE, which was defined as the 2D euclidean distance between the center of the PTP and the center of the VTP, superimposed on the phantom head Figure 30. In other words, we discarded the depth information in the error calculation. The justification for this is that the depth is already known, i.e. the VTP is on the surface of the head. Formally, the TRE is a simple 2D euclidean distance, that can be defined as:

$$TRE = d(PTP, VTP) = \sqrt{(PTP_x - VTP_x)^2 + (PTP_y - VTP_y)^2} \tag{5.11}$$

Figure 30 – Calculating the TRE: 2D euclidean distance between the center of the PTP, the black dot, at $(0,0)$, and the center of the superimposed VTP, the yellow dot, at $(4.0, -4.6)$. Coordinates in millimeters.



5.5.2 Registration time

The goal of this metric was to analyze how long it takes for the user to detect all the PRP. For this reason, what is measured is the time spent between the beginning of *Detect and classify the PRP* step (section 5.4.2) and the end of *Registration* step (section 5.4.4). In order to precisely measure this, ARNeuro saves the timestamp as soon as the registration screen appears on the device. Then, when the app detects the last PRP, the registration is performed (section 5.4.4), and a second timestamp is registered. The difference between these two timestamps is the registration time.

5.5.3 Registration RMSE

This metric is not used to measure the system accuracy. Instead, the purpose of measuring the registration RMSE is to provide a feedback to the neurosurgeon about the placement of the PRP. For example, if the value of the registration RMSE is high, it means that the neurosurgeon probably put one or more PRP in the wrong place(s) on the patient's head. This also could mean that the system is not working properly.

Formally, the registration RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} d(a_i, v_i)^2}{N}} \tag{5.12}$$

where N is the number of PRP, d is the 3D euclidean distance function, a_i is one ARRP point, and v_i is the VRP point that is "mapped" to a_i. Currently, $N = 3$, but this may change in future versions.

6

Results and discussion

In this chapter we show the results of the experiments described in section 5.5. First we explain the differences between the experiments, and then we present and discuss the results. The data sets for both experiments are in Appendix A and Appendix B.

6.1 Experiments

6.1.1 Experiment 01

The first experiment was done with a version of ARNeuro that stands between Old ARNeuro and Current ARNeuro. The reason for this in-between version is because we decided to update from ARKit 1.0 to ARKit 1.5 beta as soon as possible, since the newer version of the framework has a better capability of recognizing irregular surfaces.

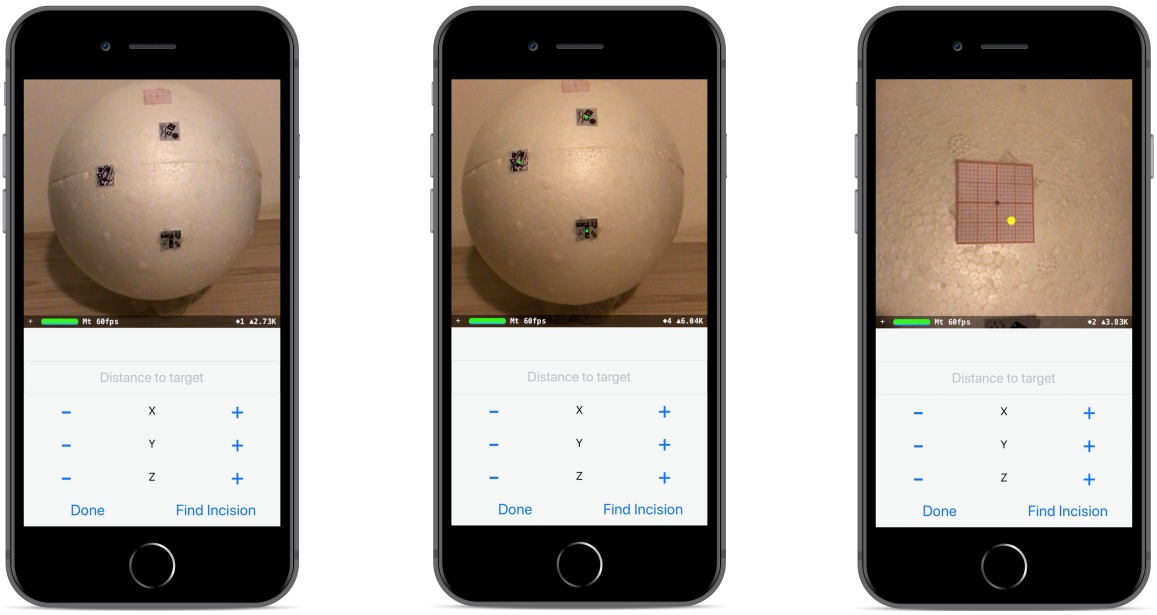
As explained in section 5.4.2.2, the image segmentation technique did not work with ARKit 1.5, so this version did not have an automatic method to recognize the PRP. This means that the user had to tap the device screen on the PRP position, which would call a *hitTest* function to create an ARRP for that PRP. Then the user had to adjust the ARRP manually, using the $-$ and $+$ buttons shown in Figure 31, because this procedure did not create the good ARRP points. This manual adjustment took some time, and this is the reason that the registration times in this experiment were higher than in Experiment 02.

Similarly to Old ARNeuro, this version still uses the trilateration method to find the center of the incision region, and it does not use any ARImageAnchor.

In this experiment, a styrofoam sphere was used to emulate the human cranium, and two different incision locations were emulated: one in temporal lobe, and another in frontal lobe. Each configuration was tested 30 times. All the tests were performed with an iPhone 7 Plus, with iOS 11.3 beta installed. It was measured accuracy (TRE), and registration time.

The main purpose of Experiment 01 was to validate both our hypotheses (section 1.5). Our tests show that it is possible to find the incision location of a craniotomy using only a mobile device to execute the Computer Vision algorithms, and also that the registration time can be shorter than the ones of the *Affordable* solutions.

Figure 31 – Experiment 01 ARNeuro. The − and + buttons were used to perform the fine adjustment of the ARRP. **(a)**: Initial screen. The user needed to tap the location of the PRP to create an ARRP. **(b)**: The ARRP, drawn in green, after the manual adjustment performed by the user. **(c)**: The VTP, drawn in yellow, found by the trilateration.



(a) (b) (c)

6.1.2 Experiment 02

The second experiment was done with Current ARNeuro. In this version, ARKit 1.5 was already released, and we used the image recognition feature of the framework. In order to avoid the behavior caused by ARImageAnchor, it was decided to apply the technique described in section 5.4.3.2 to substitute ARImageAnchor objects with ARAnchor objects.

In this experiment, a 3D-printed phantom head, based on a real patient, was used. We also had access to a CT image of the phantom head with some fiducials attached, so that we could precisely identify some landmarks on the CT image and their respective locations on the physical head. Using the software Horos, we marked 5 points on the CT image, and their coordinates can be seen in Table 7. Visual representation of the points are shown in Figure 32.

Table 7 – Coordinates, in millimeters, of the points identified in the CT image of the phantom head using the software Horos.

Name	x	y	z	Color
Point 01	37.682	-30.220	93.250	Green
Point 02	-2.908	13.955	128.250	Red
Point 03	43.947	5.795	110.750	Yellow
Point 04	75.318	2.463	48.250	Blue
Point 05	64.613	47.786	98.250	Magenta

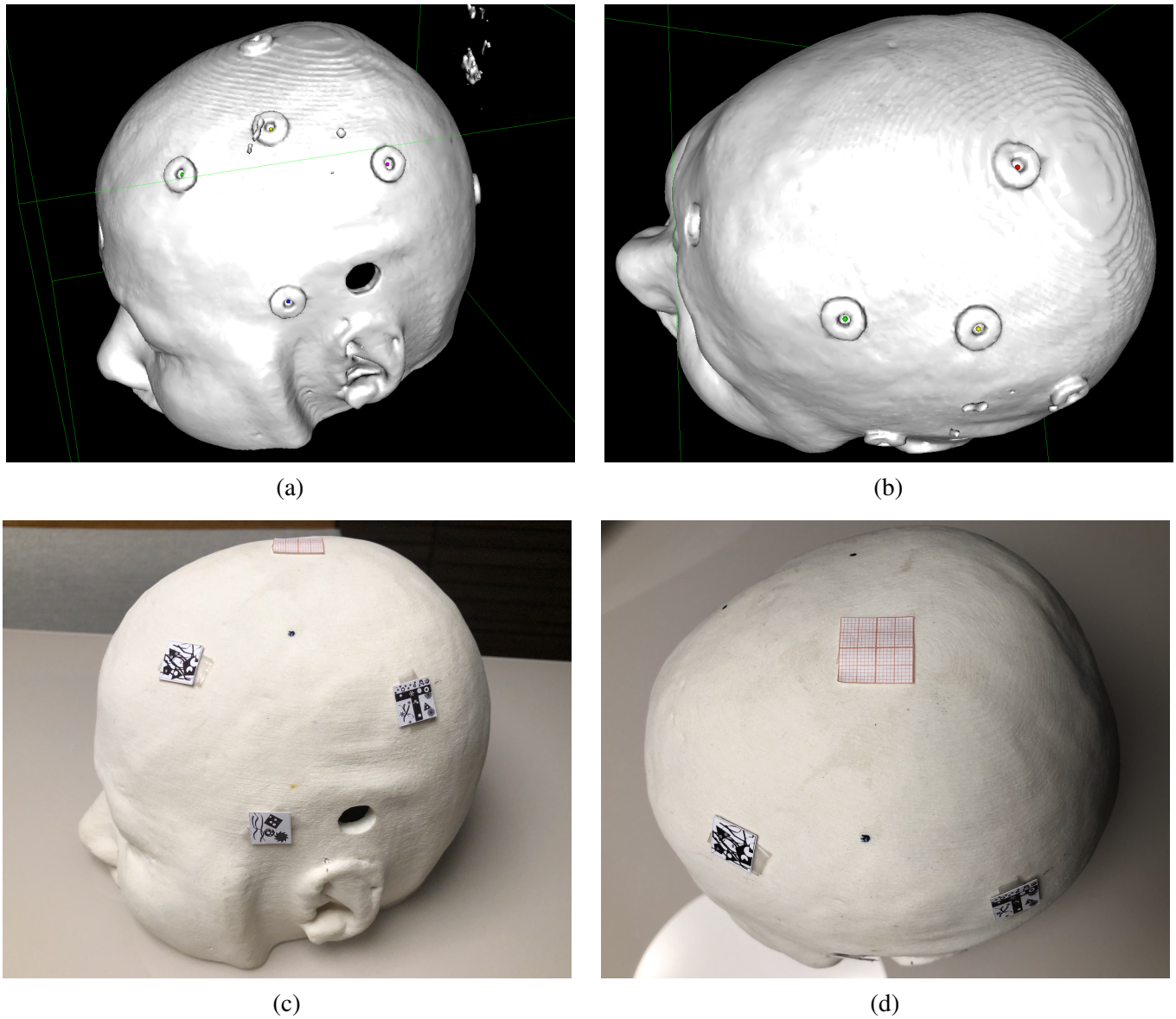
In order to verify if the CT image was an accurate representation of the physical phantom, we did a small test: (a) calculated the distances of all pairs of virtual registration points (VRP); (b) measured the distances of the pairs of physical registration points (PRP) using a caliper; (c) compared the respective distances. The the results of this test are shown in Table 8. Since all the differences between the physical distances and their virtual counterparts were smaller than 1mm, we assume that the points were correctly marked, and that the CT image is an accurate representation of the phantom head.

Table 8 – Distances, in millimeters, of each pair of points. **V** stands for the distance of virtual points, while **P** indicates the distance of physical points.

	Point 01	**Point 02**	**Point 03**	**Point 04**	**Point 05**
Point 01	-	V: 69.45 P: 69.00	V: 40.53 P: 39.80	V: 67.15 P: 66.80	V: 82.68 P: 82.55
Point 02	-	-	V: 50.68 P: 50.70	V: 112.48 P: 112.8	V: 81.26 P: 82.20
Point 03	-	-	-	V: 70.01 P: 69.90	V: 48.44 P: 48.45
Point 04	-	-	-	-	V: 68.33 P: 67.90

The PRP were indicated on the surface of the phantom head, and their correspondent VRP were marked on the CT image. See Figure 32. The emulated incision location was on the frontal lobe (Point 2). Two iPhone models, 6S Plus and 7 Plus, were tested. Both devices had iOS 11.3.4 installed, and were tested under the same conditions. Each device was tested 34 times with the same registration and target points. The 2 worst and 2 best results were discarded. It was measured accuracy (TRE), registration time, and registration RMSE.

Figure 32 – Reference points. **(a)**, **(b)**: Points marked on the CT image (VRP). **(c)**, **(d)**: Image markers placed on PRP 01, 04, and 05. The PTP, indicated by the millimeter paper, was placed on point 02.



(a) (b)

(c) (d)

6.2 Accuracy

The results of the Target Registration Error (TRE) measurements in both experiments are shown in Figure 33 and Table 9. A visual representation of the Virtual Target Point (VTP) distribution can be seen in Figure 34.

Figure 33 – Box plots of the TRE in both experiments. Values in millimeters.

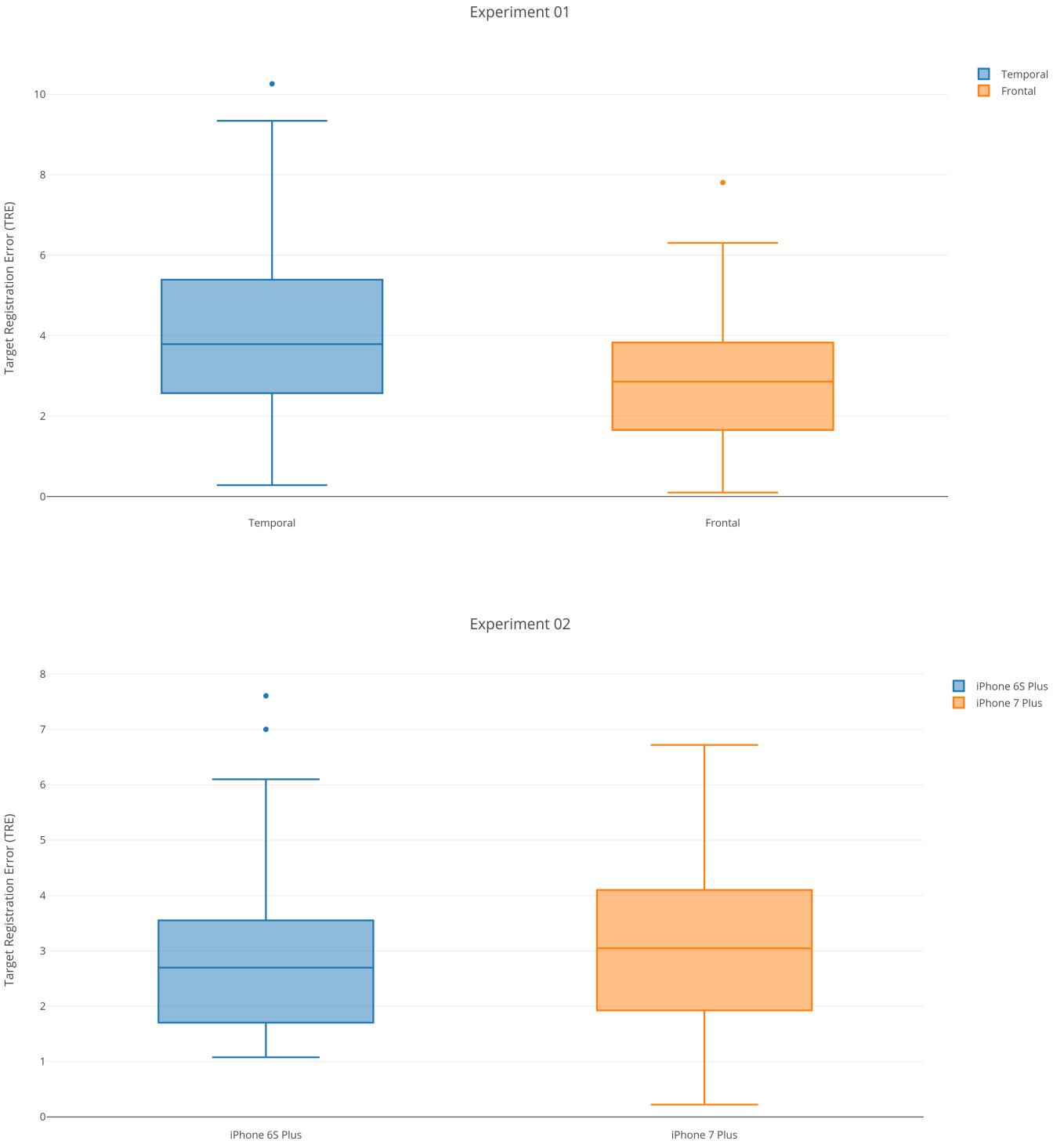
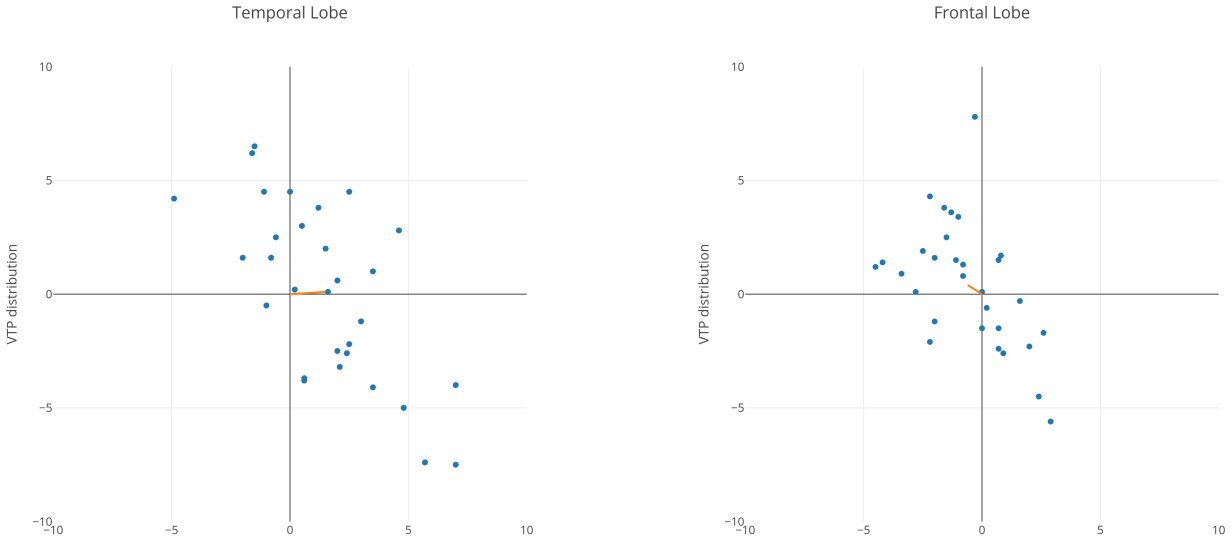


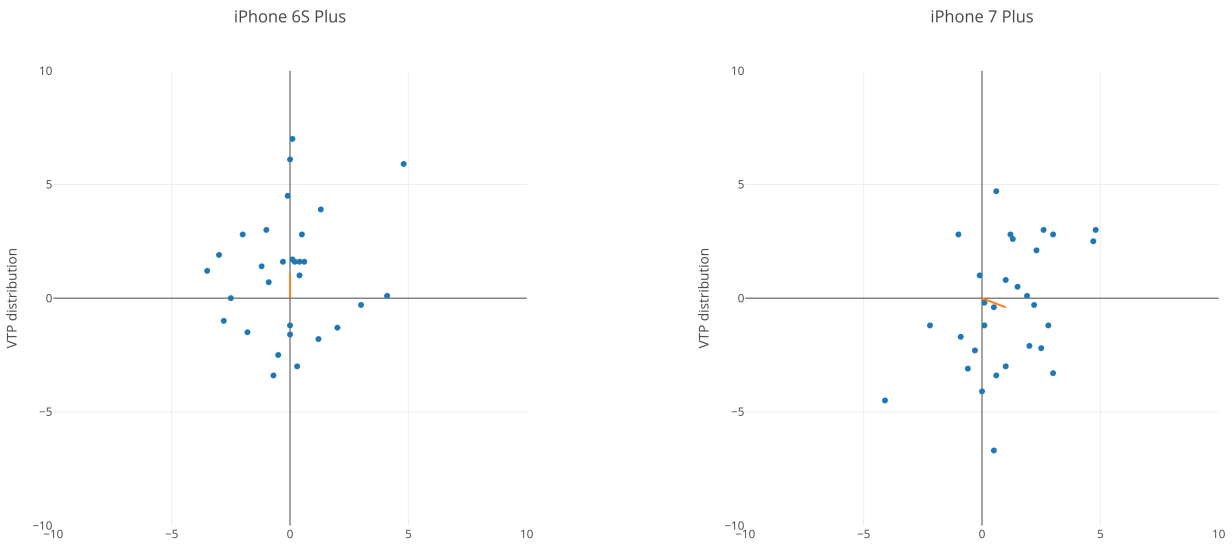
Table 9 – Summary of TRE measurements. Values in millimeters.

	Experiment 01		**Experiment 02**	
	Temporal	**Frontal**	**iPhone 6S Plus**	**iPhone 7 Plus**
Mean	4.303	2.997	2.989	3.102
Standard Deviation	2.314	1.634	1.607	1.566
Min	0.283	0.1	1.077	0.224
Max	10.259	7.806	7.606	6.719
First Quartile	2.571	1.655	1.703	1.924
Median	3.789	2.859	2.697	3.046
Third Quartile	5.391	3.828	3.551	4.100
Variance	5.354	2.669	2.581	2.453
Standard Error	0.422	0.298	0.293	0.286

Figure 34 – ARNeuro VTP distribution. The dimensions are equivalent to the millimeter paper, from (-10,-10) to (10,10). The center (0,0) is the location of the PTP. The blue dots are the VTP results of each test. The orange line represents the mean vector of all the VTP, indicating the error trend.



(a) Experiment 01



(b) Experiment 02

These results show that the non-functional requirement of accuracy (mean TRE $<$ 5mm) was achieved. In Table 10, it is possible to see how ARNeuro compares to the systems presented in chapter 2. To calculate ARNeuro mean TRE and standard deviation, we chose to use only results from Experiment 02, and to combine them. A Student's t-test indicated that both populations of Experiment 02 are not statistically different ($p = 0.788765$). Experiment 02 was chosen for the following reasons: (a) uses the current version of ARNeuro; (b) the points coordinates are obtained from the CT of a phantom head, which is a better emulation of a real situation than Experiment 01.

Table 10 – TRE comparison of ARNeuro with other AR systems. In order to present a fair comparison, if multiple results were available, we chose to include the mean TRE from tests done with phantom heads instead of with real patients. Values in millimeters.

Solution	**Mean TRE**	**Group**
Chang et al. (2012)	2.2	Substitute
Deng et al. (2014)	1.6	
Watanabe et al. (2016)	~1.0	
Léger et al. (2017)	1.76	
Maruyama et al. (2018)	3.1 ± 1.9	
Mahvash & Tabrizi (2013)	0.3	Affordable
Tabrizi & Mahvash (2015)	0.8 ± 0.25	
Eftekhar (2016)	10.2 ± 2.0	
Hou et al. (2016)	~5.0	
Chen et al. (2017)	4.4 ± 1.1	
Current ARNeuro (Experiment 02)	3.05 ± 1.59	

As it can be seen from Table 10, ARNeuro achieves better accuracy than most of the *Affordable* systems, while it is worse than most *Substitute* systems. The only system of the *Affordable* group that performs better than ARNeuro is the one that uses a video projector, introduced by Mahvash & Tabrizi (2013) and further tested by Tabrizi & Mahvash (2015). Some approaches to improve ARNeuro accuracy are discussed in section 7.1.

Since there is a lot that can be done to improve ARNeuro accuracy, the experiments results indicate that our system is a promising tool. In spite of that, it is important to make some observations:

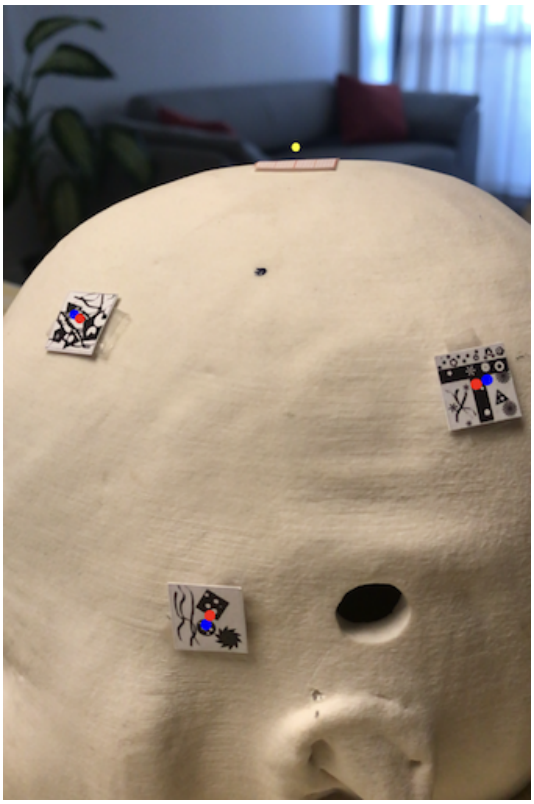
- As mentioned in section 2.4.1, the calculation of the TRE is different for each work. For this reason, the mean TRE gives a general idea of the system accuracy, but, even in systems with small errors, e.g. (MAHVASH; TABRIZI, 2013), other factors, e.g. parallax effect, may impact the software usage.

- As mentioned in section 5.5.1, the calculation of ARNeuro TRE did not include the depth. This means that the mean TRE would be higher if the depth was included. Some solutions shown in Table 10 did not include the depth either. The reasons for not including depth in the equation are: (a) we did not have a precise method to measure the depth, (b) the depth is already known, as the VTP is the surface of the head. In spite of that, if the depth of the VTP is really off, ARNeuro will suffer from a strong parallax effect, which means that the VTP perception would be heavily influenced by the device position, and thus it would not be reliable. Currently, the only way to mitigate this issue is to visually verify whether the depth is close to the head's surface or not. This is shown in Figure 35. If the depth is incorrect, especially if the point is rendered farther from the camera as it should be, e.g. inside the head, it is recommended to repeat the registration.

Our approach has many sources of error, but it is still possible to minimize the overall error. The error sources are addressed in section 6.5, and some ideas to reduce the error are presented in section 7.1. Some

examples of ARNeuro good and bad results, emphasizing the depth issue, are demonstrated in Figure 35.

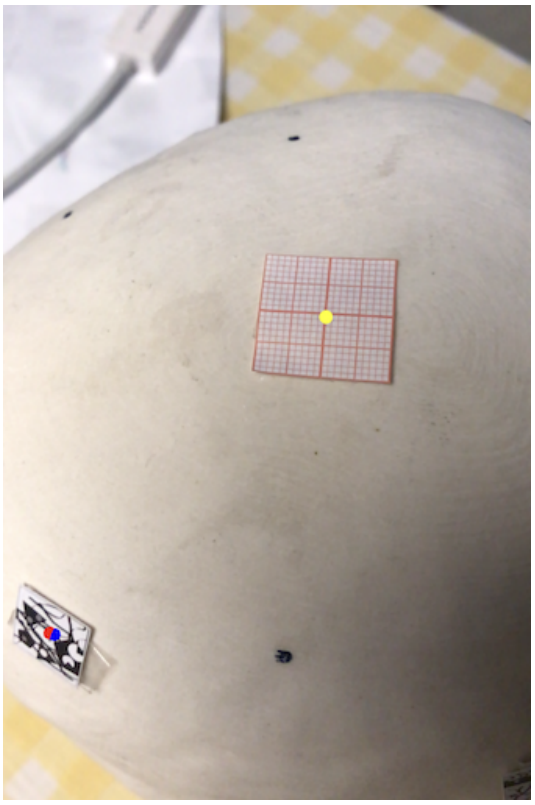
Figure 35 – Examples of ARNeuro results. **(a)**: The registration is not so good, i.e. the blue spheres are not very well aligned with the red spheres, and the depth is also not correct. **(b)**: The registration is good, but the depth is worse than it was in (a) because the parallax effect will be stronger if the VTP is located inside the head. **(c)**, **(d)**: Both were taken from the same test. The registration is good, and the location of the VTP is near perfect.



(a)

(b)

(c)



(d)

6.3 Registration time

The registration time was measured to determine whether our system is going to be faster to use than the current alternatives. The results can be seen in Figure 36 and Table 11.

Figure 36 – Box plots of the registration times in both experiments. Values in seconds.

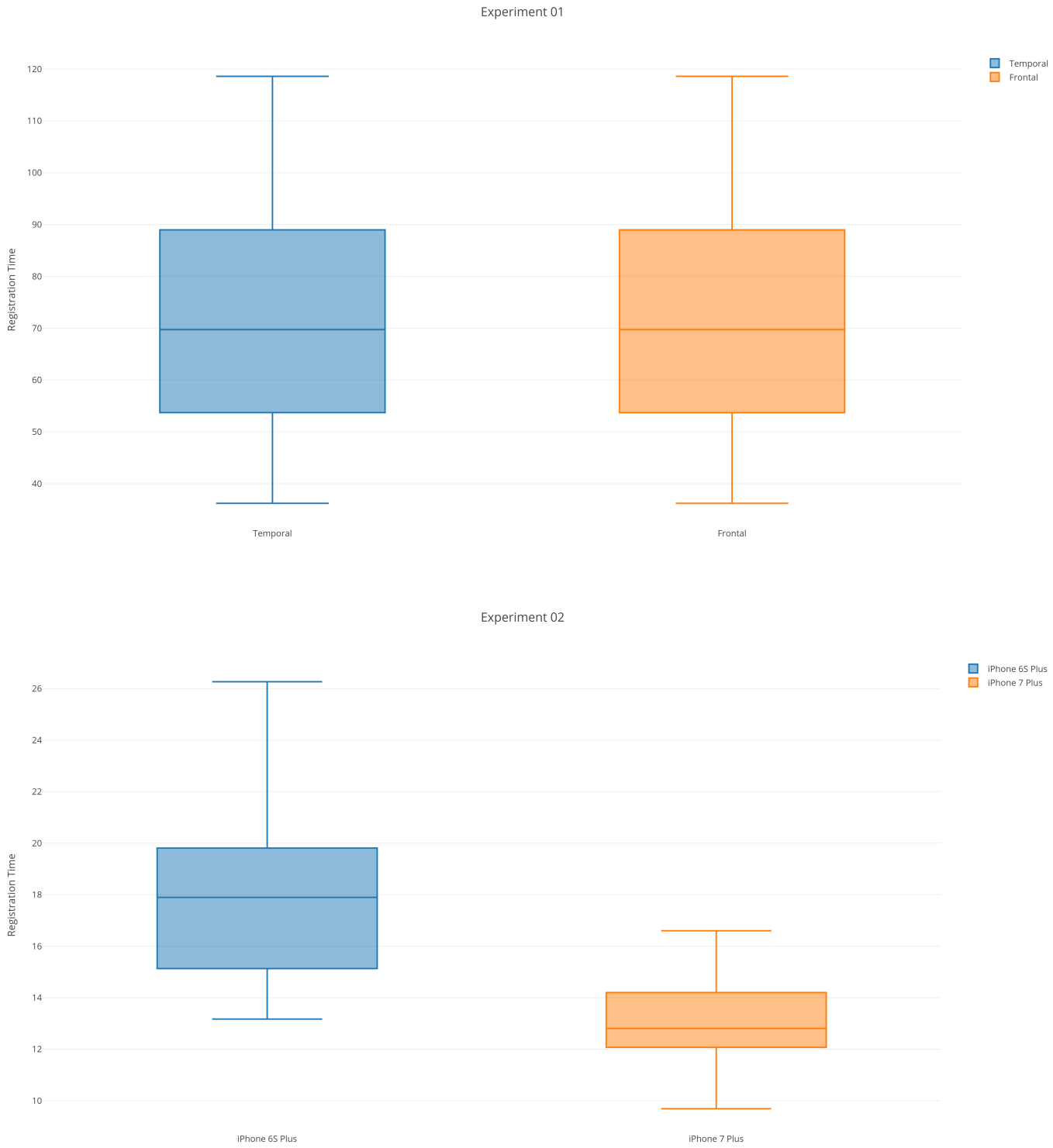


Table 11 – Summary of registration time measurements. Results are in seconds.

	Experiment 01		**Experiment 02**	
	Temporal	**Frontal**	**iPhone 6S Plus**	**iPhone 7 Plus**
Mean	71.58	53.31	17.81	12.99
Standard Deviation	20.87	18.32	3.00	1.67
Min	36.22	28.91	13.17	9.69
Max	118.6	91.18	26.27	16.6
First Quartile	53.71	38.03	15.13	12.07
Median	69.73	48.76	17.90	12.81
Third Quartile	88.98	65.29	19.81	14.2
Variance	435.79	335.65	9.03	2.78
Standard Error	3.81	3.34	0.55	0.30

From these results, it is clear that the image recognition greatly reduced the registration time. Experiment 01 ARNeuro required some manual adjustments from the user, and this is the reason it took much more time than Current ARNeuro to perform the registration.

We also compared our results with the solutions found in literature. See Table 12. As the TRE comparison, it was decided to use Experiment 02 as the reference because Current ARNeuro was tested. We did not combine the results of iPhone 6S Plus and iPhone 7 Plus tests for the comparison, because the iPhone 7 was superior. For this reason, we chose to use the values obtained from the iPhone 6S Plus tests in the comparison table.

Table 12 – Registration time comparison of ARNeuro with other AR systems. In order to present a fair comparison, if multiple results were available, we chose to include the registration time from tests done with phantom heads instead of with real patients. Values in seconds.

Solution	**Mean registration time**	**Group**
Chang et al. (2012)	–	Substitute
Deng et al. (2014)	–	
Watanabe et al. (2016)	180	
Léger et al. (2017)	–	
Maruyama et al. (2018)	180	
Mahvash & Tabrizi (2013)	300	Affordable
Tabrizi & Mahvash (2015)	228	
Eftekhar (2016)	240 ± 60	
Hou et al. (2016)	600	
Chen et al. (2017)	141.7 ± 39	
Current ARNeuro (iPhone 6S Plus)	17.81 ± 3.00	

Comparing our results with the ones of the systems found in literature, ARNeuro is clearly the best in this metric. Even our worst result from Experiment 01 (Maximum registration time = 118.6 seconds) is still better than the best mean result (141 seconds) of the works in our systematic review. It is important to note that some systems do not need a registration step, because they use some equipment with markers in predefined positions, but many of these systems need a camera calibration step, which ARNeuro does not need. ARNeuro camera calibration is performed by ARKit, while the app is being used. Another observation is that, in order to make a fair comparison, we considered only the registration time, and not the total time required by the approach. For instance, Hou et al. (2016)'s solution requires many preoperative steps that are not accounted in the registration time. Most of the analyzed papers do not mention the overall time. Also, marking the VRP on the virtual image and informing them to ARNeuro can be done before the surgery, so it should not be accounted in the registration time.

6.4 Registration RMSE

As mentioned in section 5.5.3, the main reason for measuring the registration RMSE is to provide the neurosurgeon with a feedback about the locations of the three PRP. If the markers locations are correct, then this value should be low. The registration RMSE was measured only in Experiment 02. Since our tests were performed with the markers in ideal positions, i.e. errors introduced by the markers positions were low, the RMSE was also low. The results can be seen in Figure 37 and section 5.5.3.

Figure 37 – Box plots of the RMSE of the distances between the VRP and the ARRP. Values in millimeters.

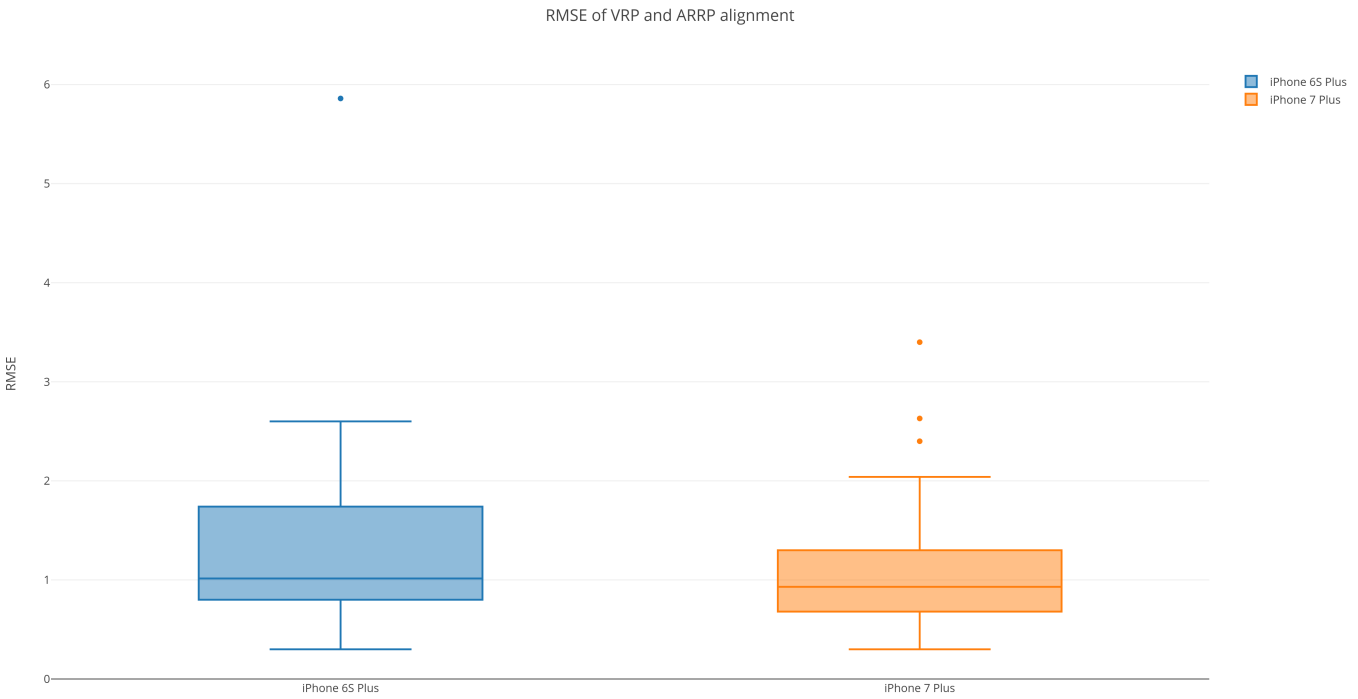


Table 13 – Summary of RMSE of the distances between VRP and ARRP. Results are in millimeters.

	iPhone 6S Plus	**iPhone 7 Plus**
Mean	1.36	1.13
Standard Deviation	1.02	0.70
Min	0.3	0.3
Max	5.86	3.4
First Quartile	0.8	0.68
Median	1.02	0.93
Third Quartile	1.74	1.3
Variance	1.04	0.49
Standard Error	0.19	0.13

The results indicate that the PRP were placed in correct positions, which were determined by the phantom head fiducials and the phantom head CT. This also means that, if the surgeon uses a CT of the patient with fiducials as a reference, the registration should give good results, i.e. low registration RMSE.

Since this metric is very similar to the Fiducial Registration Error (FRE), it is crucial to make an important observation: a low registration RMSE does not imply that the TRE will be good. There is a common belief that the FRE is a good indicator of the TRE, but it is not, as it was explained by Fitzpatrick (2010), and Labadie (2016). In fact, we present some examples that corroborate those studies. As it is shown in Table 14, it is possible to have a high registration RMSE and low TRE, as well as low registration RMSE and high TRE.

Table 14 – Analysis of the relationship between Registration RMSE and TRE. Values in millimeters.

Device	Trial	Registration RMSE	TRE
iPhone 6S Plus	09	0.97	6.100
	11	5.86	2.973
iPhone 7 Plus	14	2.04	6.088
	22	3.40	4.738
	25	0.53	4.104

As explained in section 5.5.3, we conclude that the registration RMSE is a good metric to check if the PRP were correctly placed and the system is working properly, but it is not a good accuracy indicator. This metric may be more useful in tests where the precise position of the PRP is not known. For example, when the surgeon uses craniometric landmarks as references. Since this approach is more prone to error, this metric should provide more interesting results.

6.5 Limitations

This work has two kinds of limitations: (a) the research limitations, which are related to the scope and tests of work, and (b) the application limitations, which are ARNeuro characteristics that could be improved. In this section, we detail all the limitations that we could think. In section 7.1, we propose future work to handle these limitations.

6.5.1 Research

The main limitation of our research was the broad scope. A direct consequence of this was our test methodology. The focus of our tests was to verify if the incision location of a craniotomy could be found, with great accuracy and in a short period of time, using only a mobile device. Both of these hypotheses, presented in chapter 1, were proven true in our experiments. In spite of that, ARNeuro is a system that involves many parts, e.g. hardware, software, human interaction, etc., and a variety of different tests could be performed for each of these variables.

Regarding our experiments, Experiment 01 was done before including more advanced features of ARKit, e.g. image recognition, but we decided to include our findings to enrich the discussion about the application improvements. It is possible to see from the results that the registration time was greatly reduced in Experiment 02. The accuracy, on the other hand, shows little difference between the versions. A big limitation of our experiments is that they were not performed in a very controlled environment. For example, we did not measure the lighting conditions or the electromagnetic interference, which are two factors that can impact ARNeuro. Also, the experiments were not performed in the operating room and with real patients.

Finally, we did not conduct deep investigations about the effects of the parallax caused by an error of the VTP depth, and about the minimum acceptable registration RMSE. These two factors should be further examined to ensure an even fairer comparison with other systems.

6.5.2 Application

The main limitation of ARNeuro is its heavy dependence on ARKit, which is a closed source framework. Here we comment all of the reasons that we could think this dependence can be an issue:

- **ARKit is an AR framework built for general augmented reality applications**

  It is clear that the developers of ARKit have concerns regarding its accuracy, but, as the framework was designed for mobile devices and for the development of general AR applications, e.g. games, it is expected some trade-offs between accuracy and the usage of the device resources. In other words, the developers may opt for less accuracy in order to improve the device power consumption, and the use of CPU or GPU. In fact, the metric scale of ARKit is "meter", and not "millimeter", which may indicate that the framework main applications are not concerned with errors in the millimeter scale.

- **It is not possible to write a good debug or change the code**

  ARKit is a closed source framework, so it is difficult to write a good debug, and it is impossible to change the internal code. This means that we are left with the only option of black-box testing, so we make assumptions about what the framework is doing based on its behavior. Furthermore, since the code cannot be modified, we cannot create our own optimizations in the internal code.

- **Changes in the framework may have an impact on the application**

  The development of ARKit is very active. The framework was released in May, 2017 and two big updates were released until June, 2018. The problem is that these updates may change the behavior of ARKit and, consequently, change the behavior of ARNeuro.

Now that the main limitation was detailed, we argument why ARKit is still a good option. First, even though the developers may have opted to reduce accuracy in order to make a better use of the device resources, we showed in our experiments that a good accuracy is achievable, and we also discuss some ideas of improvements in section 7.1. Second, ARKit allows that the developer writes custom code to execute in response to some events in the form of delegates. For instance, every time a new frame arrive from the camera, or every time the anchors are updated, ARKit informs the application trough delegate functions. This mitigates the issue of not being able to change the code, and it also helps in writing debugs. Third, to alleviate the impact of ARKit updates, one solution is to rely on the most basic functionalities of the framework: its VIO Slam, and the *hitTest* function. It is almost safe to assume that these two functionalities will only improve with future releases. Since this assumption is not guaranteed, another important measure would be to create an accuracy test to validate whether ARNeuro is behaving properly or not. Finally, ARKit was especially useful in this work because we could show, in a short time frame, that it is possible to have an accurate "small neuronavigator" in a mobile device. Developing from scratch a complete and accurate VIO Slam approach that works well on commercial mobile devices is an effort of many software engineers.

In a user level, ARNeuro has limitations concerning usability and accuracy. Since only the registration time was measured as a usability metric, the only limitation that we can think in terms of pure usability is what was stated in previous work: the device obstructs a clear working space, because it stays between the surgeon and the patient (EFTEKHAR, 2016). We did not consider the "small screen" an issue because ARNeuro may be installed in tablets. Also, our solution does not have the lag mentioned by many authors, because the application does not need an external computer. A study with emphasis on user experience may be done in the future, when the application is ready for clinical use.

Regarding the application accuracy, ARNeuro has many sources of error that can affect the location of the VTP. The correct positioning of the VTP leads to a low TRE, and, consequently, high accuracy. Thus, we mention all the factors that we could think that impacts the VTP location.

- **Environment lighting**

  Dark environments make difficult for the camera to see the details, therefore the VIO is unable to locate the features in the image. This should not be a problem in a clinical environment.

- **Device motion**

  If the device is moved too fast or the user shakes it, the measurements provided by the sensors to the VIO will not be reliable. This is something that can be easily avoided by the surgeon or the assistant that handles the device.

- **Scene observed by the camera**

  If the scene does not contain interesting regions, i.e. features, the VIO cannot perform the triangulation to extract scene information. An example of a scene without features would be a blank wall. This should not be a problem in a surgery, because the patient's head should have enough features, including the image markers themselves.

- **Time of use**

  Error is accumulated over time, so this can contribute to decrease the accuracy. As explained in section 3.3, this error is reduced by the Visual Odometry of ARKit. This also should not be a problem in a clinical environment, and it was not a problem in our tests. Also, if the environment knowledge is updated by the framework, the application receives a notification and then we can inform the user that a new registration has to be performed.

- **Distance from the device to the target**

  Moving the device far away from the target also can affect the accuracy, because ARKit may internally update its virtual map to best fit the other parts of the environment. In our experiments, the image markers were identified from a distance of approximately 4cm. ARKit should work well in a range of 1.5m, so this also should not be a problem in clinical use.

- **Battery health and low power mode**

  VIO Slam requires a lot from CPU and GPU. Apple devices have a mechanism to protect the phone if the battery is old, reducing the CPU power. Low power mode also reduces the CPU power. Therefore, to achieve best results, ARNeuro should be executed when the battery is healthy, and low power mode should not be active.

- **Device temperature**

  If the device gets too hot, the IMU sensors may be affected, and the measurements they provide may contain more errors.

- **External interference**

  Electromagnetic sources may have some impact on the device sensors. To avoid this issue in clinical use, some tests can be performed to verify the readings of the magnetometer and determine whether it is safe to use the application or not.

- **Sensors and factory calibration**

  The quality of the sensors in the device and how well they are calibrated can contribute to the error. In fact, this was the problem that happened with the iPhone 8 (section 6.5.3).

- **Virtual image errors**

  The virtual image, which is usually obtained from a CT/MRI scan, should be correct, otherwise it will be impossible to perform a correct registration.

- **Medical imaging viewer software errors**

  For the same reason as the previous item, the image viewer software should correctly inform the coordinates of the selected points on the virtual image.

- **ARNeuro implementation errors**

  Incorrect implementations of the techniques used in ARNeuro may affect the software accuracy. An error in the registration step, for instance, will directly impact the location of the VTP.

- **Selection of the points in the virtual image**

  The points have to be precisely localized in the virtual image, or ARNeuro will work with wrong coordinates. This could be improved by implementing an algorithm that does this automatically, instead of relying on manual selection of the points.

- **PRP positioning**

  The PRP have to be precisely positioned on the patient's head, otherwise the registration will not produce good results.

As it can be seen, although ARNeuro has many sources of error, a good accuracy was achieved in our experiments. Furthermore, in section 7.1, we propose some approaches that may reduce the error even more. The main goals of discussing these sources of error are to serve as a guide on what to avoid while using the app, and to understand what can be improved to optimize the system accuracy.

6.5.3 iPhone 8 tests

In Experiment 02, we also had an iPhone 8 Plus available. After many tests with this device, it was decided to remove it from our experiments. The reason is that the iPhone 8 Plus was showing a systematic error in which the VTP would never be rendered in the correct place. Actually, the VTP was almost always rendered out of the millimeter paper area. Since we had good results with the iPhone 6S Plus, and with the iPhone 7 Plus, we opted to test ARNeuro in another iPhone 8 device, different from the one we had. The systematic error was also present in the tests with the other iPhone 8, so we concluded that this should be an error of ARKit 1.5. For this reason, we filed a bug report with Apple (ID# 43583932). The developers contacted us and suggested to test our application with ARKit 2.0, in order to see if the error persists. Since ARKit 2.0 is currently in beta, and it requires a beta version of the iOS, this test was not performed yet. For now, we conclude that Current ARNeuro, with ARKit 1.5, is not reliable on the iPhone 8 or iPhone 8 Plus.

7

Conclusion and future work

More than 200,000 people were diagnosed with brain and nervous system cancer worldwide in 2012. More than 11,000 new cases are expected in Brazil this year. In some cases, the survival rate is low, and it is important to provide adequate treatment. Image-guidance has been of great importance in neurosurgical procedures, assisting in minimally invasive surgeries, which leads to less pain, shorter hospital stays, and faster recoveries. In this context, neuronavigators became important tools for the neurosurgeon. In spite of that, traditional neuronavigators still have some limitations, and researchers are proposing to use AR in order to overcome some of these limitations. Many AR systems have been developed and tested by research groups. Some of these systems aim to replace the traditional neuronavigator, while others just focus on providing an affordable solution, which could be used in hospitals with less resources. The problem is that the *Substitute* solutions are expensive and hard to adopt, and the *Affordable* solutions require many preoperative steps from the neurosurgeon.

In order to verify the state-of-the-art approaches presented in literature, a systematic review was done. The search was designed to be broad, returning as many results as possible related to AR and neurosurgery. On the other hand, the inclusion and exclusion parameters were very specific, to leave only relevant work for analysis. Quantitative and qualitative analyses were performed with the selected papers. The advantages and limitations of the systems are discussed, improved by the complementary information of two previous systematic reviews. The specialist Bruno Fernandes de Oliveira Santos, MD, MSc, also validated our review based on his personal experience.

We propose a new approach to assist in craniotomy planning. ARNeuro is a standalone mobile AR application that uses VIO Slam to extract scene information and find the center of the incision region of the craniotomy. Comparing to the *Affordable* and *Substitute* solutions mentioned in the Introduction, ARNeuro requires little information and setup, making it more usable than the *Affordable* solutions. Our system is also affordable and easy to adopt, making it a good alternative to neuronavigators and *Substitute* solutions in a hospital with few resources. Also, ARNeuro could be an alternative to these more complete solutions when only craniotomy localization is important.

Two versions of ARNeuro were tested, and we have shown that both our hypotheses were true. The results of our experiments show that ARNeuro accuracy is not as good as that of neuronavigators and other AR systems. In spite of that, according to the literature, the accuracy is considered good, and our approach provides better results than traditional techniques for craniotomy localization. The limitations have to be kept in mind

when using the system, but most of them can be informed to the user during the application execution.

Finally, there is much work to be done. ARNeuro seems to be a promising system, but it is not ready for clinical use yet. An important observation from our experiments is that more tests are needed, in a more controlled environment. A deeper investigation has to be done to identify the sources and the behavior of the TRE, so the accuracy can be improved through the implementation of new techniques. Also, it is crucial to develop a better method to measure the TRE, since the parallax effect caused by a wrong depth in the VTP may affect the user perception of the incision region. With the accuracy improved, it would be possible to create a complete neuronavigation system: the first standalone mobile neuronavigation system. Currently, the application already calculates the map between the ICS and the WCS, which means that loading a CT/MRI to the device would be enough to create this navigation system, as the application would be able to superimpose the virtual image on the patient's head.

7.1 Future work

In this work, we proved it is possible to have a simple neuronavigation system that executes on a commercial mobile device. Unfortunately, the current version of the system is still not ready for clinical use. In this section, we propose some possible future work.

- **Perform experiments in a more controlled environment**

  It is necessary to test ARNeuro in a more controlled environment to understand what factors contribute most to the error. Also, it is important to create a new method to calculate the TRE, taking in consideration the 3D position of the VTP, and to evaluate the impact of the parallax effect caused by a wrong depth of the VTP.

- **Automate the method to get the VRP**

  Currently, the method to mark the VRP in the CT/MRI involves a manual procedure performed by the user. This is not only time consuming, but also is a source of error, because it is difficult to precisely mark the locations of the VRP. An automatic method to process the virtual image file would solve this issue.

- **Implement an easier method to pass the VRP and the VTP to the mobile device**

  The first step of ARNeuro requires that the user types the coordinates of the points. This does not take much time, but it could be done in an easier way, like reading a QR code.

- **Use more points for the VTP**

  In its current version, ARNeuro requires only one point as the VTP because typing too many points would take time. If an easier method to get the points is implemented, it is better to use more points to represent the VTP, as a complete incision region could be represented instead of just its center.

- **Test other registration methods**

  In their study, Mascott et al. (2006) show the results of different registration methods, and the best approach used 5 fiducial registration points. Many other methods could be implemented in ARNeuro, e.g. fiducial registration with more than 3 points, surface matching, cranial landmarks, and even facial recognition.

- **Test other frameworks**

  Our experiments were performed with ARKit 1.5 but there are many other frameworks that can be used to test the same approach, e.g. Vuforia, ARCore, Wikitude, and even ARKit 2.0.

- **Improve ARRP positioning**

  ARNeuro estimates the positions of the ARRP based on the image recognition, but sometimes the estimated point is millimeters off the correct place. There are some ways to improve the ARRP positioning. One option would be to use image tracking instead of just image detection. The first continuously tracks the image, while the latter just detects the image but does not track it. ARKit 2.0 and Vuforia have image tracking, for example.

- **Perform clinical studies**

  ARNeuro should be tested in clinical environments and with patients in order to prove that our method can be used in surgeries.

- **Implement an automatic method to test the device sensors and the environment conditions**

  It is important to develop a method to check if ARNeuro can be used. The main reason for this is because, as it was shown in chapter 6, some devices, e.g. iPhone 8, may show some errors that will impact the application accuracy. Also, some environments may not be suitable for the use of ARNeuro, e.g. an environment with high electromagnetic interference.

- **Develop a specific VIO**

  ARNeuro is very dependent on ARKit. One option to solve this limitation is to develop a specific VIO algorithm. This would also make possible to perform optimizations to the scenario of of medical use.

- **Develop a complete neuronavigator**

  ARNeuro uses the VIO Slam of ARKit to perform the device tracking and to create the internal representation of the physical environment. To be a complete AR neuronavigator, the only missing thing is to render the CT/MRI superimposed on the patient's head.

- **Use the app for training**

  ARNeuro can be tested for training new surgeons. It would be useful to evaluate whether the system improves the learning experience or not.

- **Test usability**

  Qualitative experiments to verify if ARNeuro improves the usability over traditional neuronavigators and other AR methods would provide a good feedback of whether this method can replace a neuronavigator in the future.

- **Calibrate and certificate ARNeuro as a medical instrument**

  We have shown that it is possible to provide AR neuronavigation in a mobile device, but this is not enough to make ARNeuro a commercial product. It is important to calibrate and certificate ARNeuro as a medical application, but this requires a deeper study to determine if we can use the current solution or if it is needed to implement another approach.

Bibliography

ABE, Y. et al. A novel 3d guidance system using augmented reality for percutaneous vertebroplasty. *Journal of Neurosurgery: Spine*, American Association of Neurological Surgeons, v. 19, n. 4, p. 492–501, 2013. 2 citations on pages 28 and 29.

ABHARI, K. et al. Training for planning tumour resection: augmented reality and human factors. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 62, n. 6, p. 1466–1477, 2015. 2 citations on pages 23 and 28.

ABHARI, K. et al. Use of a mixed-reality system to improve the planning of brain tumour resections: preliminary results. In: SPRINGER. *Workshop on Augmented Environments for Computer-Assisted Interventions*. [S.l.], 2012. p. 55–66. One citation on page 20.

ALARAJ, A. et al. Role of cranial and spinal virtual and augmented reality simulation using immersive touch modules in neurosurgical training. *Neurosurgery*, Oxford University Press, v. 72, n. suppl_1, p. A115–A123, 2013. One citation on page 28.

AMERICAN CANCER SOCIETY, INC. *Survival Rates for Selected Adult Brain and Spinal Cord Tumors*. 2017. Website. Available at:<https://www.cancer.org/cancer/brain-spinal-cord-tumors-adults/detection-diagnosis-staging/survival-rates.html>. Last accessed 07 June 2018. One citation on page 19.

APPLE INC. *iOS Drawing Concepts*. 2012. Website. Available at:<https://developer.apple.com/library/archive/documentation/2DDrawing/Conceptual/DrawingPrintingiOS/GraphicsDrawingOverview/GraphicsDrawingOverview.html>. Last accessed 27 August 2018. One citation on page 58.

APPLE INC. *ARKit Documentation*. 2018. Website. Available at:<https://developer.apple.com/documentation/arkit/>. Last accessed 17 June 2018. 3 citations on pages 43, 45, and 48.

APPLE INC. *Understanding ARKit Tracking and Detection*. 2018. Video and slide presentation. Available at:<https://developer.apple.com/videos/play/wwdc2018/610/>. Last accessed 17 June 2018. 4 citations on pages 43, 46, 47, and 59.

ARUN, K. S.; HUANG, T. S.; BLOSTEIN, S. D. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, n. 5, p. 698–700, 1987. 2 citations on pages 63 and 64.

AZIMI, E.; DOSWELL, J.; KAZANZIDES, P. Augmented reality goggles with an integrated tracking system for navigation in neurosurgery. In: IEEE. *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. [S.l.], 2012. p. 123–124. One citation on page 23.

AZUMA, R. T. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, MIT Press, v. 6, n. 4, p. 355–385, 1997. One citation on page 22.

BISSON, M.; CHERIET, F.; PARENT, S. 3d visualization tool for minimally invasive discectomy assistance. *Studies in health technology and informatics*, v. 158, p. 55–60, 2010. One citation on page 29.

BRAINLAB AG. *Microscope Navigation*. 2018. Website. Available at:<https://www.brainlab.com/surgery-products/overview-neurosurgery-products/microscope-navigation/>. Last accessed 13 June 2018. One citation on page 20.

CABRILO, I.; SCHALLER, K.; BIJLENGA, P. Augmented reality-assisted bypass surgery: embracing minimal invasiveness. *World neurosurgery*, Elsevier, v. 83, n. 4, p. 596–602, 2015. One citation on page 28.

CHANG, Y.-Z. et al. Application of real-time single camera slam technology for image-guided targeting in neurosurgery. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Applications of Digital Image Processing XXXV*. [S.l.], 2012. v. 8499, p. 84992D. 7 citations on pages 19, 23, 31, 32, 35, 73, and 76.

CHEN, J.-g. et al. Presurgical planning for supratentorial lesions with free slicer software and sina app. *World neurosurgery*, Elsevier, v. 106, p. 193–197, 2017. 9 citations on pages 19, 22, 23, 32, 33, 35, 36, 73, and 76.

CUTOLO, F. et al. A new head-mounted display-based augmented reality system in neurosurgical oncology: a study on phantom. *Computer Assisted Surgery*, Taylor & Francis, v. 22, n. 1, p. 39–53, 2017. 10 citations on pages 9, 19, 22, 23, 32, 33, 35, 36, 39, and 40.

DAS, M. et al. Augmented reality visualization for ct-guided interventions: system description, feasibility, and initial evaluation in an abdominal phantom. *Radiology*, Radiological Society of North America, v. 240, n. 1, p. 230–235, 2006. One citation on page 29.

DENG, W. et al. Easy-to-use augmented reality neuronavigation using a wireless tablet pc. *Stereotactic and functional neurosurgery*, Karger Publishers, v. 92, n. 1, p. 17–24, 2014. 7 citations on pages 23, 31, 32, 34, 35, 73, and 76.

DIBIASE, D. *The Nature of Geographic Information*. [S.l.]: John E. Dutton e-Education Institute, Penn State University, 2013. One citation on page 64.

DROUIN, S. et al. Ibis: an or ready open-source platform for image-guided neurosurgery. *International journal of computer assisted radiology and surgery*, Springer, v. 12, n. 3, p. 363–378, 2017. 3 citations on pages 20, 22, and 23.

DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, IEEE, v. 13, n. 2, p. 99–110, 2006. One citation on page 47.

DWARAKANATH, S. et al. Neuronavigation in a developing country: a pilot study of efficacy and limitations in intracranial surgery. *Neurology India*, Medknow, v. 55, n. 2, p. 111, 2007. One citation on page 22.

EFTEKHAR, B. A smartphone app to assist scalp localization of superficial supratentorial lesions. *World neurosurgery*, Elsevier, v. 85, p. 359–363, 2016. 10 citations on pages 20, 22, 23, 32, 35, 36, 40, 73, 76, and 79.

ENCHEV, Y. Neuronavigation: geneology, reality, and prospects. *Neurosurgical focus*, American Association of Neurological Surgeons, v. 27, n. 3, p. E11, 2009. One citation on page 21.

ESKENAZI, M. *Three things marketers should know about adopting augmented reality*. 2018. Website. Available at:<http://www.thedrum.com/opinion/2018/03/29/three-things-marketers-should-know-about-adopting-augmented-reality>. Last accessed 17 June 2018. One citation on page 46.

FERLAY, J. et al. Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. *International journal of cancer*, Wiley Online Library, v. 136, n. 5, 2015. One citation on page 19.

FERRARI, V.; CUTOLO, F. Letter to the editor: Augmented reality–guided neurosurgery. v. 125, p. 1–2, 05 2016. One citation on page 36.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Readings in computer vision*. [S.l.]: Elsevier, 1987. p. 726–740. One citation on page 46.

FITZPATRICK, J. M. The role of registration in accurate surgical guidance. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, SAGE Publications Sage UK: London, England, v. 224, n. 5, p. 607–622, 2010. One citation on page 77.

FRIGHETTO, L. et al. Image-guided frameless stereotactic biopsy sampling of parasellar lesions. technical note. *Journal of neurosurgery*, v. 98, n. 4, p. 920–925, 2003. One citation on page 19.

FRITZ, J. et al. Mr-guided vertebroplasty with augmented reality image overlay navigation. *Cardiovascular and interventional radiology*, Springer, v. 37, n. 6, p. 1589–1596, 2014. 2 citations on pages 23 and 29.

GAVAGHAN, K. et al. Evaluation of a portable image overlay projector for the visualisation of surgical navigation data: phantom studies. *International journal of computer assisted radiology and surgery*, Springer, v. 7, n. 4, p. 547–556, 2012. One citation on page 23.

GERARD, I. J. et al. Improving patient specific neurosurgical models with intraoperative ultrasound and augmented reality visualizations in a neuronavigation environment. In: SPRINGER. *Workshop on Clinical Image-Based Procedures*. [S.l.], 2015. p. 28–35. One citation on page 20.

GERARD, I. J. et al. Combining intraoperative ultrasound brain shift correction and augmented reality visualizations: a pilot study of eight cases. *Journal of Medical Imaging*, International Society for Optics and Photonics, v. 5, n. 2, p. 021210, 2018. 2 citations on pages 21 and 23.

GILDENBERG, P. L.; LABUZ, J. Use of a volumetric target for image-guided surgery. *Neurosurgery*, Oxford University Press, v. 59, n. 3, p. 651–659, 2006. 2 citations on pages 23 and 29.

GOOGLE INC. *Building an augmented reality (AR) application using the WebXR Device API*. 2018. Website. Available at:<https://codelabs.developers.google.com/codelabs/ar-with-webxr/>. Last accessed 17 June 2018. One citation on page 60.

GOOGLE LLC. *ARCore Documentation*. 2018. Website. Available at:<https://developers.google.com/ar/>. Last accessed 17 June 2018. One citation on page 44.

GUHA, D. et al. Augmented reality in neurosurgery: a review of current concepts and emerging applications. *Canadian Journal of Neurological Sciences*, Cambridge University Press, v. 44, n. 3, p. 235–245, 2017. 8 citations on pages 25, 27, 28, 31, 33, 39, 41, and 50.

HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. [S.l.]: Cambridge university press, 2003. One citation on page 46.

HOCEVAR, S. *Fast branchless RGB to HSV conversion in GLSL*. 2013. Website. Available at:<http://lolengine.net/blog/2013/07/27/rgb-to-hsv-in-glsl>. Last accessed 27 August 2018. One citation on page 56.

HOROS PROJECT. *Horos*. 2018. Website. Available at:<https://horosproject.org/>. Last accessed 17 June 2018. 4 citations on pages 9, 44, 49, and 51.

HOU, Y. et al. A low-cost iphone-assisted augmented reality solution for the localization of intracranial lesions. *PloS one*, Public Library of Science, v. 11, n. 7, p. e0159185, 2016. 14 citations on pages 9, 19, 20, 22, 23, 31, 32, 33, 35, 36, 37, 49, 73, and 76.

INOUE, D. et al. Preliminary study on the clinical application of augmented reality neuronavigation. *Journal of Neurological Surgery Part A: Central European Neurosurgery*, Georg Thieme Verlag KG, v. 74, n. 02, p. 071–076, 2013. 4 citations on pages 20, 22, 23, and 29.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. *Estimativa 2018: incidência de câncer no Brasil*. 2017. Website. Available at:<http://www.inca.gov.br/estimativa/2018/estimativa-2018.pdf>. Last accessed 07 June 2018. One citation on page 19.

KANTELHARDT, S. R. et al. Video-assisted navigation for adjustment of image-guidance accuracy to slight brain shift. *Operative Neurosurgery*, Oxford University Press, v. 11, n. 4, p. 504–511, 2015. One citation on page 29.

KAWAMATA, T. et al. Endoscopic augmented reality navigation system for endonasal transsphenoidal surgery to treat pituitary tumors. *Neurosurgery*, Oxford University Press, v. 50, n. 6, p. 1393–1397, 2002. One citation on page 23.

KERSTEN-OERTEL, M. et al. Augmented reality in neurovascular surgery: feasibility and first uses in the operating room. *International journal of computer assisted radiology and surgery*, Springer, v. 10, n. 11, p. 1823–1836, 2015. 2 citations on pages 22 and 23.

KERSTEN-OERTEL, M. et al. Augmented reality for specific neurovascular surgical tasks. In: SPRINGER. *Workshop on Augmented Environments for Computer-Assisted Interventions*. [S.l.], 2015. p. 92–103. One citation on page 28.

KERSTEN-OERTEL, M. et al. Towards augmented reality guided craniotomy planning in tumour resections. In: SPRINGER. *International Conference on Medical Imaging and Virtual Reality*. [S.l.], 2016. p. 163–174. One citation on page 28.

KOCKRO, R. A. et al. Dex-ray: augmented reality neurosurgical navigation with a handheld video probe. *Neurosurgery*, Oxford University Press, v. 65, n. 4, p. 795–808, 2009. 2 citations on pages 20 and 23.

LABADIE, R. F. *Image-guided surgery: fundamentals and clinical applications in otolaryngology*. [S.l.]: Plural Publishing, 2016. One citation on page 77.

LÉGER, É. et al. Quantifying attention shifts in augmented reality image-guided neurosurgery. *Healthcare technology letters*, IET, v. 4, n. 5, p. 188–192, 2017. 13 citations on pages 9, 19, 22, 23, 24, 27, 32, 35, 38, 39, 41, 73, and 76.

LEUTENEGGER, S. et al. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, SAGE Publications Sage UK: London, England, v. 34, n. 3, p. 314–334, 2015. One citation on page 46.

LIAO, H. et al. 3-d augmented reality for mri-guided surgery using integral videography autostereoscopic image overlay. *IEEE transactions on biomedical engineering*, IEEE, v. 57, n. 6, p. 1476–1486, 2010. One citation on page 23.

LOVO, E. E. et al. A novel, inexpensive method of image coregistration for applications in image-guided surgery using augmented reality. *Operative Neurosurgery*, Oxford University Press, v. 60, n. suppl_4, p. ONS–366, 2007. One citation on page 22.

LOW, D. et al. Augmented reality neurosurgical planning and navigation for surgical excision of parasagittal, falcine and convexity meningiomas. *British journal of neurosurgery*, Taylor & Francis, v. 24, n. 1, p. 69–74, 2010. One citation on page 23.

MACIUNAS, R. J. Computer-assisted neurosurgery. *Clinical neurosurgery*, LIPPINCOTT WILLIAMS & WILKINS, v. 53, p. 267, 2006. One citation on page 20.

MAHVASH, M.; TABRIZI, L. B. A novel augmented reality system of image projection for image-guided neurosurgery. *Acta neurochirurgica*, Springer, v. 155, n. 5, p. 943–947, 2013. 6 citations on pages 23, 32, 35, 36, 73, and 76.

MANDEL, M. et al. 3d preoperative planning in the er with osirix®: when there is no time for neuronavigation. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 13, n. 5, p. 6477–6491, 2013. One citation on page 19.

MARUYAMA, K. et al. Smart glasses for neurosurgical navigation by augmented reality. *Operative Neurosurgery*, 2018. 9 citations on pages 9, 23, 32, 35, 36, 39, 40, 73, and 76.

MASCOTT, C. R. et al. Quantification of true in vivo (application) accuracy in cranial image-guided surgery: influence of mode of patient registration. *Operative Neurosurgery*, Oxford University Press, v. 59, n. suppl_1, p. ONS–146, 2006. 3 citations on pages 20, 33, and 83.

MEOLA, A. et al. Augmented reality in neurosurgery: a systematic review. *Neurosurgical review*, Springer, v. 40, n. 4, p. 537–548, 2017. 13 citations on pages 9, 12, 19, 22, 23, 27, 28, 31, 33, 34, 35, 36, and 41.

MIESNIEKS, M. *Why is ARKit better than the alternatives?* 2017. Website. Available at:<https://medium.com/6d-ai/why-is-arkit-better-than-the-alternatives-af8871889d6a>. Last accessed 17 June 2018. 3 citations on pages 43, 46, and 47.

NAVAB, N.; HEINING, S.-M.; TRAUB, J. Camera augmented mobile c-arm (camc): calibration, accuracy study, and clinical applications. *IEEE transactions on medical imaging*, IEEE, v. 29, n. 7, p. 1412–1423, 2010. One citation on page 29.

NGUYEN, H. *Gpu gems 3*. [S.l.]: Addison-Wesley Professional, 2007. One citation on page 57.

NISTÉR, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. In: IEEE. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. [S.l.], 2004. v. 1, p. I–I. One citation on page 46.

PALEOLOGOS, T. S. et al. Clinical utility and cost-effectiveness of interactive image-guided craniotomy: clinical comparison between conventional and image-guided meningioma surgery. *Neurosurgery*, Oxford University Press, v. 47, n. 1, p. 40–48, 2000. One citation on page 19.

PTC INC. *Vuforia*. 2018. Website. Available at:<https://www.vuforia.com/>. Last accessed 17 June 2018. One citation on page 44.

ROBERTS, D. W. et al. A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope. *Journal of neurosurgery*, Journal of Neurosurgery Publishing Group, v. 65, n. 4, p. 545–549, 1986. One citation on page 27.

SCHROEDER, H. W. et al. Frameless neuronavigation in intracranial endoscopic neurosurgery. *Journal of neurosurgery*, Journal of Neurosurgery Publishing Group, v. 94, n. 1, p. 72–79, 2001. One citation on page 20.

SHENAI, M. B. et al. Virtual interactive presence and augmented reality (vipar) for remote surgical assistance. *Operative Neurosurgery*, Oxford University Press, v. 68, n. suppl_1, p. ons200–ons207, 2011. One citation on page 29.

SPIVAK, C. J.; PIROUZMAND, F. Comparison of the reliability of brain lesion localization when using traditional and stereotactic image-guided techniques: a prospective study. *Journal of neurosurgery*, Journal of Neurosurgery Publishing Group, v. 103, n. 3, p. 424–427, 2005. 3 citations on pages 20, 23, and 26.

STADIE, A. T. et al. Neurosurgical craniotomy localization using a virtual reality planning system versus intraoperative image–guided navigation. *International journal of computer assisted radiology and surgery*, Springer, v. 6, n. 5, p. 565–572, 2011. One citation on page 23.

TABRIZI, L. B.; MAHVASH, M. Augmented reality–guided neurosurgery: accuracy and intraoperative application of an image projection technique. *Journal of neurosurgery*, American Association of Neurological Surgeons, v. 123, n. 1, p. 206–211, 2015. 11 citations on pages 9, 19, 20, 23, 31, 32, 35, 36, 37, 73, and 76.

TAGAYTAYAN, R.; KELEMEN, A.; SIK-LANYI, C. Augmented reality in neurosurgery. *Archives of Medical Science*, Termedia, v. 12, n. 1, 2016. One citation on page 28.

WACKER, F. K. et al. An augmented reality system for mr image–guided needle biopsy: initial results in a swine model. *Radiology*, Radiological Society of North America, v. 238, n. 2, p. 497–504, 2006. One citation on page 29.

WAGNER, W. et al. Cranial neuronavigation in neurosurgery: assessment of usefulness in relation to type and site of pathology in 284 patients. *min-Minimally Invasive Neurosurgery*, Georg Thieme Verlag Stuttgart· New York, v. 43, n. 03, p. 124–131, 2000. One citation on page 23.

WATANABE, E. et al. The trans-visible navigator: a see-through neuronavigation system using augmented reality. *World neurosurgery*, Elsevier, v. 87, p. 399–405, 2016. 14 citations on pages 9, 19, 22, 23, 28, 31, 32, 35, 36, 38, 39, 40, 73, and 76.

WEISS, C. R. et al. Augmented reality visualization using image-overlay for mr-guided interventions: system description, feasibility, and initial evaluation in a spine phantom. *American Journal of Roentgenology*, Am Roentgen Ray Soc, v. 196, n. 3, p. W305–W307, 2011. One citation on page 29.

WESARG, S. et al. Accuracy of needle implantation in brachytherapy using a medical ar system: a phantom study. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*. [S.l.], 2004. v. 5367, p. 341–353. One citation on page 23.

WIKITUDE GMBH. *Wikitude*. 2018. Website. Available at:<https://www.wikitude.com/>. Last accessed 17 June 2018. One citation on page 44.

WILES, A. D. et al. A statistical model for point-based target registration error with anisotropic fiducial localizer error. *IEEE transactions on medical imaging*, IEEE, v. 27, n. 3, p. 378–390, 2008. One citation on page 25.

WOERDEMAN, P. A. et al. Application accuracy in frameless image-guided neurosurgery: a comparison study of three patient-to-image registration methods. *Journal of neurosurgery*, American Association of Neurological Surgeons, v. 106, n. 6, p. 1012–1016, 2007. One citation on page 33.

WU, J.-R. et al. Real-time advanced spinal surgery via visible patient model and augmented reality system. *Computer methods and programs in biomedicine*, Elsevier, v. 113, n. 3, p. 869–881, 2014. 2 citations on pages 28 and 29.

XIA, Z. et al. Minimally invasive surgery is superior to conventional craniotomy in patients with spontaneous supratentorial intracerebral hemorrhage: A systematic review and meta-analysis. *World neurosurgery*, Elsevier, v. 115, p. 266–273, 2018. One citation on page 19.

ZAMBONI, A. et al. Start uma ferramenta computacional de apoio à revisão sistemática. In: *Congresso Brasileiro de Software (CBSoft'10). Salvador, Brazil*. [S.l.: s.n.], 2010. One citation on page 29.

Appendix

APPENDIX A – Experiment 01 Data Set

	Temporal			Frontal		
Trial	**dx**	**dy**	**Time**	**dx**	**dy**	**Time**
01	-1.0	-0.5	63.18	0.2	-0.6	77.28
02	0.5	3.0	58.74	-0.3	7.8	68.74
03	1.2	3.8	101.23	0.7	1.5	53.54
04	7.0	-7.5	88.98	-1.3	3.6	88.36
05	0.0	4.5	92.91	-2.2	-2.1	56.34
06	-1.5	6.5	118.60	2.0	-2.3	74.25
07	-1.1	4.5	72.04	-1.6	3.8	49.99
08	2.5	-2.2	79.40	-4.2	1.4	32.17
09	2.1	-3.2	75.19	-2.2	4.3	34.23
10	7.0	-4.0	67.42	-1.1	1.5	57.45
11	4.6	2.8	99.86	2.4	-4.5	31.42
12	3.5	1.0	94.06	2.9	-5.6	43.17
13	3.0	-1.2	43.97	1.6	-0.3	51.40
14	4.8	-5.0	87.12	0.7	-2.4	45.87
15	-2.0	1.6	45.75	-2.5	1.9	81.49
16	1.5	2.0	51.35	-2.8	0.1	33.70
17	5.7	-7.4	81.06	-1.5	2.5	28.91
18	-1.6	6.2	92.12	-4.5	1.2	51.30
19	-0.8	1.6	66.46	0.0	0.1	53.90
20	2.0	-2.5	65.94	-3.4	0.9	36.63
21	-0.6	2.5	53.42	-2.0	1.6	38.03
22	0.2	0.2	36.22	0.7	-1.5	33.85
23	0.6	-3.7	100.08	-1.0	3.4	47.52
24	2.0	0.6	72.41	0.0	-1.5	43.76
25	1.6	0.1	62.94	2.6	-1.7	90.65
26	2.5	4.5	47.85	-0.8	0.8	44.69
27	-4.9	4.2	55.65	0.8	1.7	93.18
28	2.4	-2.6	36.79	0.9	-2.6	65.29
29	3.5	-4.1	53.71	-2.0	-1.2	46.14
30	0.6	-3.8	82.96	-0.8	1.3	45.95

The coordinates dx and dy, relative to the origin, i.e. the center of the millimeter paper, are in millimeters. The times are in seconds.

APPENDIX B – Experiment 02 Data Set

	iPhone 6S Plus				iPhone 7 Plus			
Trial	**dx**	**dy**	**Time**	**Reg. RMSE**	**dx**	**dy**	**Time**	**Reg. RMSE**
01	-0.1	4.5	16.43	0.89	4.8	3.0	10.03	0.42
02	-0.3	1.6	14.41	1.14	1.0	-3.0	13.35	1.09
03	0.0	-1.6	13.17	0.78	1.3	2.6	15.12	1.43
04	-0.9	0.7	14.11	0.88	2.3	2.1	13.96	0.77
05	-1.0	3.0	18.09	1.13	-1.0	2.8	14.20	1.73
06	2.0	-1.3	16.09	0.30	0.5	-0.4	12.16	0.39
07	0.0	-1.2	16.05	0.60	2.8	-1.2	14.75	1.05
08	0.1	1.7	21.10	1.31	-0.3	-2.3	12.05	0.68
09	0.0	6.1	18.24	0.97	2.5	-2.2	13.24	2.63
10	0.2	1.6	13.80	1.60	0.5	-6.7	16.60	0.81
11	-2.8	-1.0	19.98	5.86	0.1	-0.2	16.30	0.49
12	-0.7	-3.4	16.28	1.74	1.5	0.5	14.73	0.88
13	4.8	5.9	15.08	2.37	2.6	3.0	13.18	0.69
14	1.2	-1.8	13.62	0.50	-4.1	-4.5	12.57	2.04
15	-2.0	2.8	17.77	2.08	0.6	-3.4	12.96	1.30
16	-3.0	1.9	18.02	1.87	-0.1	1.0	12.31	1.19
17	-1.2	1.4	21.21	0.98	1.0	0.8	12.07	0.84
18	-1.8	-1.5	26.27	0.58	1.9	0.1	12.14	0.61
19	0.1	7.0	18.70	2.60	-2.2	-1.2	11.09	2.40
20	0.4	1.6	19.86	1.59	2.2	-0.3	9.69	0.86
21	-0.5	-2.5	23.22	1.05	4.7	2.5	11.09	0.98
22	0.3	-3	19.79	0.89	0.6	4.7	13.18	3.40
23	1.3	3.9	17.71	2.19	-0.6	-3.1	12.65	0.30
24	-2.5	0.0	19.14	0.80	-0.9	-1.7	14.80	0.57
25	3.0	-0.3	14.30	0.64	3.0	2.8	11.52	0.53
26	0.6	1.6	15.13	1.14	0.0	-4.1	11.06	1.80
27	4.1	0.1	19.11	0.78	2.0	-2.1	15.27	1.13
28	0.5	2.8	19.81	1.90	1.2	2.8	12.61	0.83
29	-3.5	1.2	20.67	0.82	3.0	-3.3	13.12	1.11
30	0.4	1.0	17.19	0.89	0.1	-1.2	12.14	1.09

The coordinates dx and dy, relative to the origin, i.e. the center of the millimeter paper, are in millimeters. The registration RMSE is also in millimeters. The times are in seconds.